

DX Skill Landscape & Timeline - 2026-08-25

Table of Contents

Introduction	16
Timelines	16
Coverage for IPA DSS-P	17
Principles	18
Primary References	18
01 - Business Transformation	19
Project Management	19
Project Planning & Estimation	19
Prioritization	20
Work & Issue Tracking	20
Team Collaboration	21
Process Maturity & Quality Standards	22
Product Management	22
Product Strategy & Roadmapping	22
Requirements Analysis	23
Marketing & Customer Experience	23
Marketing Technology	24
Metrics & Performance	26
Goal Setting Frameworks	26
Product & Customer Metrics	27
Engineering Performance Metrics	27
Enterprise Strategy & Architecture	27
Enterprise Architecture	27
Enterprise Applications	28
Organizational Change Management	29
Strategic Analysis & Business Intelligence	30
Risk Management	30
IT Service Management	31
Knowledge & Content Management	31
Personal Knowledge Management	32
Collaborative Workspaces & Wikis	32
Content Management Systems	33

Content Collaboration & File Sync	33
Enterprise AI & Productivity	33
Autonomous AI Workers	33
Enterprise AI Assistants	34
Self-Hosted AI Platforms	34
AI Content & Document Generation	34
AI Agent Registries	35
Human-Centered Design	35
Core Principles & User Experience (UX)	35
User Research & Ideation	36
Cognitive & Behavioral Psychology	37
Visual Design & Typography	37
Design Systems & Design Tools	39
Web Experience & Performance	40
Economics, Game Theory & Finance	40
Economics & Game Theory	40
Corporate Finance & Markets	41
Accounting & Financial Reporting	42
02 - Web Application Development	44
Web Platform Fundamentals	44
Web Concepts	44
Browser Technologies & DOM	45
Web Application Architectures	46
Frontend Development	47
UI Frameworks & Core Libraries	47
State, Routing & Logic	49
Styling & UI Components	49
Build & Development Tooling	51
Full-stack & Static Site Frameworks	51
Full-stack Frameworks	51
Static Site Generators	52
Headless CMS	52
Backend Development	53
API Architectural Styles	53
Backend Frameworks	53
Web Infrastructure	55

Web Server & Proxy	56
CDN & Edge Computing	56
Decentralized Web	57
Blockchain Technology	58
Decentralized Social	59
Development & Testing Tools	59
Web/HTTP Clients	60
Web Debugging Tools	61
Web Test Automation Frameworks	61
03 - Cloud & Cloud-Native Computing	63
Cloud Computing	63
Computing & Storage (IaaS)	63
Networking	63
Application Hosting Platform (PaaS)	64
Cloud Command Line Interfaces	64
Cloud Emulators	64
Private Cloud & On-Premises IaaS	65
Cloud Architecture Frameworks	65
Configuration as Code	65
Infrastructure as Code (IaC)	65
AI-driven Infrastructure	65
Configuration Management & Automation	66
Image Building	66
Ecosystem & Vendor Tools	66
Containerization	67
Engines & Runtimes	68
Image Management	68
Environment & Management	69
WebAssembly	70
Kubernetes	71
Core Concepts & Components	72
Operations & Management	73
CLI & Local Environments	74
Ecosystem & Extensions	74
Cloud-Native Computing	75
Container as a Service (CaaS)	75

Function as a Service (FaaS)	76
Advanced Runtimes & Isolation	76
Cloud-Native Infrastructure	77
CI/CD & GitOps	78
Delivery & Deployment	79
GitOps & Cloud-Native	79
Integrations & Registries	80
System Observability	80
Instrumentation & Platforms	80
Telemetry Shipment	81
Telemetry Collection & Storage	82
SRE (Site Reliability Engineering)	83
Fleet Management & Operations	84
Chaos Engineering	85
FinOps	85
Performance & Load Testing	85
Performance Testing Tools	86
04 - Security & Privacy	86
Security Foundations	86
Common Threats & Attack Vectors	87
Modern Security Architectures	87
Security Training & Competitions	88
Cryptography & Data Protection	88
Core Cryptography	88
Public Key Infrastructure (PKI)	90
Secrets Management	91
Applied Cryptography & Tools	92
Identity & Access Management (IAM)	93
Integrated IAM	93
Authentication (AuthN)	94
Authorization (AuthZ)	95
Secure Development Lifecycle (DevSecOps)	96
Secure Design & Modeling	97
Secure Development Practices	97
Web Application Security	98
Application Security Testing (AST)	99

Infrastructure as Code (IaC) Security	100
Software Supply Chain Security (SSCS)	101
Runtime & Operational Security	102
Cloud-Native Application Protection (CNAPP)	102
Security Operations & Monitoring (SecOps)	103
Policy Enforcement	104
Digital Forensics & Incident Response (DFIR)	104
Secure Communications & Networking	105
Transport Layer Security (TLS)	105
Secure Shell (SSH)	105
Firewalls & Network Protection	106
Email & DNS Security	106
Governance, Risk, and Compliance (GRC)	107
Data Governance	107
AI Governance & Security	108
Regulations & Standards	108
Vulnerability Management & Reporting	109
System & Personal Security	110
OS & Endpoint Security	110
Personal Security Tools	111
05 - Data Science & Engineering	111
Foundational Concepts	111
General Data Concepts & Principles	111
Core Data Engineering & Database Concepts	112
Data Governance, Quality & Architecture	113
Data Science Toolkit	113
Programming Languages & Libraries	113
Specialized & Scientific Tools	114
Data Sources & Geospatial	114
Spreadsheet & Collaborative Data Platforms	115
Interactive Computing Environments	115
Data Visualization	115
Common Chart Types	115
Visualization Tools & Libraries	116
Dashboarding & Web Apps	117
Distributed Systems	117

Distributed Computing Principles	117
Consensus & Replication Strategies	118
Distributed Patterns & Observability	118
Distributed Storage Systems	118
Mathematics & Statistics	119
Base Mathematics	119
Probability & Information Theory	121
Statistics & Numerical Methods	121
Data Formats & Architecture	123
Data Formats & Table Formats	123
Data Architectures & Methodologies	123
Data Governance & Metadata Management	124
Data Quality & Validation	124
Data Versioning & Schema Management	124
Relational Databases (SQL)	124
SQL Fundamentals	125
Database Management Systems (DBMS)	125
Cloud & Managed Services	126
Connectivity & Tooling	127
NoSQL & Specialized Databases	128
NoSQL Data Models	128
Vector & AI Databases	129
Cloud NoSQL Services	130
Data Processing & Messaging	130
Enterprise Integration	130
Message Queuing & Event Streaming	131
Batch Processing (ETL/ELT)	131
Stream Processing	132
Data Analytics & Search	133
Search Engines & Platforms	133
Analytics Engines & Platforms	134
Semantic Layer	135
06 - AI, Machine Learning & LLM	135
Foundations of AI & Machine Learning	135
Machine Learning	137
Learning Paradigms	137

Neural Networks and Deep Learning	140
Architectures	141
Natural Language Processing (NLP)	142
Concepts & Vector Representations	142
Computer Vision	143
Image Processing & Fundamentals	144
3D Vision & Geometry	144
Object Detection & Recognition	144
Pose & Body Estimation	144
Face & Biometric Recognition	144
OCR & Text Recognition	144
Speech & Audio Processing	145
Automatic Speech Recognition (ASR)	145
Text-to-Speech (TTS) & Voice Synthesis	145
Audio Processing & Tools	145
Voice Activity Detection & Audio Enhancement	146
Speaker Identification & Voice Biometrics	146
Generative AI & Large Language Models (LLMs)	146
Model Providers & Aggregators	146
Standards & Model Formats	147
Model Interface SDKs & Clients	147
Techniques & Methods	148
Development Tools & Utilities	148
Benchmarking & Analysis	148
Agentic AI	149
Standards & Protocols	149
Agentic Patterns & Techniques	150
Agent Orchestration & Frameworks	150
Agent Development Kits (ADK)	151
Supporting Services & Platforms	152
MLOps & LLMOps	153
ML Lifecycle & MLOps Platforms	153
LLM Serving & Runtimes	154
LLMOps & Observability	154
07 - Fundamental Developer Skills	155
Software Development Methods	155

Agile Development	155
Lean Development	157
DevOps & Engineering Productivity	159
Release Automation	159
Release Conventions & Standards	159
The Open Ecosystem	160
Open Source	160
Open Data	161
Community & Governance	162
Shell & Terminal	163
Bash & Other Shells	164
Shell Utilities	164
Terminal Emulators	165
Terminal Utilities	166
Linux or Unix-like environments on Windows	167
Version Control System	167
Git Hosting Services	169
Branching Models	169
Code Review	169
Integrated Development Environment (IDE)	170
Language Servers	172
Code Quality & Refactoring	172
Analysis Platform	173
Formatters	173
Linters	174
Coding style guides	174
Developer AI & Productivity	174
AI Productivity Tools	175
Development Agents	175
Supporting Tools & Infrastructure	177
08 - OS & Network Basics	178
Core OS Concepts	178
Process Management	178
Inter-Process Communication (IPC)	179
Memory Management	179
Storage Management	179

Base Network Concepts & Protocols	180
Link Layer (L2)	180
Internet Layer (L3)	181
Transport Layer (L4)	182
Network Architectures	182
Domain Name System (DNS)	182
Domain Registration & Lookup	182
Server & Resolver Implementations	183
Client Tools	183
Email System	183
Mailbox Formats	184
Server Software (MTA/MDA)	184
Client Software & Utilities	184
Spam Test and Reputation	185
Unix-like Operating Systems	185
Linux Distributions	187
BSD Distributions	187
System Services & Auth	188
Machine Virtualization	188
Type-1 Hypervisors	188
Type-2 Hypervisors	188
CPU Emulators	189
Computer Hardware	189
Linux Host Administration	189
Core Utilities	189
Process & System Monitoring	190
Time Synchronization	191
Modern CLI Alternatives	191
Package Management Tools	192
Linux Network Administration	192
Configuration & Management	192
Analysis & Security	193
Proxies & Gateways	193
File Sharing & Remote Access	193
File Servers and Protocols	194
Remote Access Servers and Protocols	194

09 - Programming Concepts & Paradigms	194
Software Design & Architecture	194
Design Principles	195
Design Best Practices	195
Design Patterns	196
Architectural Styles	196
Architecture Description	197
Domain-Driven Design (DDD)	197
Core Programming Concepts	198
Language Mechanics & Execution	198
Memory Management	199
Control Flow Structures	199
Foundational Techniques & Properties	200
Module Structure & Organization	200
Programming Paradigms	200
Object-oriented Programming	201
Functional Programming	201
Reactive Programming & Others	202
Scripting Languages	203
Python	203
JavaScript & TypeScript	204
Ruby, Perl & Others	206
Asynchronous & Concurrency	207
Language Analysis	208
Program Translation	209
Major Compiler Infrastructures	210
Specific Translators & Build Tools	210
Linkers (Standalone)	210
Program Execution	210
Runtime Implementations	211
Algorithm & Computational Complexity	212
Algorithm	213
Data Structures	214
10 - Advanced Programming	216
Procedural & Systems Programming	216
Rust Language	216

Go Language	217
C & Other Procedural Languages	218
Functional & Hybrid Programming	218
Functional-first & Multi-paradigm Languages	218
Multi-paradigm & Hybrid Languages	219
Data & Format Standards	220
Text Format & Character Code	220
DateTime Format	220
Data Exchange Languages	221
Configuration Languages	222
Text Processing & Manipulation	223
Regular Expression	223
Text Tools	223
Tabular Data	224
Template Engines	224
Markup & Document Processing	225
Debugging	225
Logging	226
Test Frameworks & Tools	227
Test Frameworks	228
Assertion Libraries	229
Code Coverage Tools	229
Test Supporting Tools	229
Mutation Testing	230
Build Automation	230
Monorepo Management	231
Program Documentation	231
Package Dependency Management	232
Virtual Environment	233
11 - Specialized Development Domains	234
Business & Productivity Application SDKs	234
Project & Work Management	234
Collaboration & Communication	235
Cloud Storage & File Management	235
Email & Marketing Automation	235
Payments & Finance	235

CRM & Customer Support	236
Enterprise Workspaces	236
Developer Tools Integration SDKs	236
Version Control & DevOps	237
Container & Orchestration	237
CI/CD & Automation	237
Computer Graphics & Game Development	238
3D Graphics	238
2D Graphics	238
Graphics APIs	238
Binary & Media Processing	238
Binary Format Tools	238
Data Serialization	239
Image & Media Formats	239
Image & Media Processing	239
Compression & Archiving	240
Document Processing	240
Business Document Formats	241
PDF Processing	241
Office Document Processing	242
CLI/TUI Development	242
Desktop App Development	245
Client OS & Environments	245
GUI Systems & Windowing	246
Desktop GUI Toolkits	246
Desktop App Testing & Automation	247
Installation & Packaging	247
Mobile App Development	247
Mobile Platforms & Native SDKs	248
Mobile Cross-Platform Frameworks	248
Mobile DevOps & Testing	248
Application Services & Features	248
Internet of Things (IoT)	249
IoT Concepts	249
Communication Standards	250
IoT Hardware Platforms	250

IoT Cloud Platforms	251
Low-Code & No-Code Development.....	252
Business Application Platforms	252
Workflow & Integration Automation	252
Web Content & Portal Builders	252
12 - Personal Skills.....	253
Foundational Thinking & Logic	253
Logic	253
Informal Logic	253
Mathematical Logic.....	254
Foundational Concepts	254
Logical Systems.....	254
Branches of Mathematical Logic	255
Documentation	257
Architectural Documentation	257
Lightweight Markup & Writing Styles	258
Documentation Tooling	260
Interpersonal & Team Leadership	260
Team Dynamics & Communication	260
Organizational Behavior	262
Leadership Styles	262
Individual Psychology & Performance.....	263
Personal Performance	263
Social Performance.....	264
Timeline - 1930-89.....	265
1930s.....	265
1940s.....	265
1950s.....	266
1960s.....	266
1970-74.....	269
1975-79.....	271
1980-84.....	273
1985.....	276
1986.....	277
1987.....	278
1988.....	279

1989.....	281
Timeline - 1990-99.....	282
1990.....	282
1991.....	283
1992.....	284
1993.....	285
1994.....	287
1995.....	288
1996.....	290
1997.....	292
1998.....	293
1999.....	295
Timeline - 2000-09.....	297
2000.....	297
2001.....	298
2002.....	301
2003.....	302
2004.....	303
2005.....	305
2006.....	307
2007.....	309
2008.....	311
2009.....	313
Timeline - 2010-14.....	316
2010.....	316
2011.....	319
2012.....	322
2013.....	324
2014.....	327
Timeline - 2015-19.....	331
2015.....	331
2016.....	334
2017.....	337
2018.....	340
2019.....	343
Timeline - 2020-24.....	344

2020..... 344

2021..... 347

2022..... 349

2023..... 351

2024..... 354

Timeline - 2025-Present..... 356

2025..... 356

2026..... 360

Introduction

This site aims to provide a comprehensive overview of concepts, techniques, tools, platforms, and frameworks related to DX (Digital Transformation), featuring Software Development and DevOps spaces, spanning the following 12 categories:

- [01 - Business Transformation](#)
- [02 - Web Application Development](#)
- [03 - Cloud & Cloud-Native Computing](#)
- [04 - Security & Privacy](#)
- [05 - Data Science & Engineering](#)
- [06 - AI, Machine Learning & LLM](#)
- [07 - Fundamental Developer Skills](#)
- [08 - OS & Network Basics](#)
- [09 - Programming Concepts & Paradigms](#)
- [10 - Advanced Programming](#)
- [11 - Specialized Development Domains](#)
- [12 - Personal Skills](#)

Timelines

In addition to the skill categories above, this site provides [chronological timelines](#) documenting significant technology events, product releases, and industry milestones from 1930 to the present. Each entry is categorized by domain using emojis:

- 🏢 Business Administration, Development Methodology, Management
- 🧠 UX, Design & Cognitive Science
- 🌐 Web Technology incl. Web Frameworks
- ☁ Cloud, Cloud Native, Container, DevOps/SRE
- 🛡 Security, Privacy
- 🦋 Malware, Virus, Security Incident
- 📊 Data Science, Databases, Data Platforms
- 🧠 AI, Machine Learning, Large Language Models
- 💻 Shell, Terminal, IDE, Developer Productivity
- ⚙ System Administration, OS, VM, Network Infrastructure

-  Programming Paradigms, Programming Concepts, Libraries
-  Others

These timelines help trace the evolution of technologies listed in the skill sections and provide historical context for understanding current trends in software development and digital transformation.

Coverage for IPA DSS-P

This site's skill list attempts to cover all of the **Digital Skills Standard for Promotion (DSS-P) v2.0** defined by IPA. Under each subsection of each section page, the most relevant DSS-P skill names are listed.

The IPA (Information-technology Promotion Agency, Japan) is a policy implementation agency under the jurisdiction of Japan's Ministry of Economy, Trade and Industry (METI). It plays a central role in Japan's national IT strategy, including **Human Resource Development** (administering national IT examinations), **Information Security Measures**, and establishing **Guidelines for Digital Transformation (DX)**.

The **DSS-P** referenced in this document is a public standard defined by the IPA to accelerate DX in Japanese companies. It is widely adopted by many Japanese enterprises as a benchmark for talent development and hiring.

The table below shows the primary DSS-P category covered by each skill section:

Section - Title	Primary DSS-P Category
01 - Business Transformation	1. Business Transformation
02 - Web Application Development	3. Technology
03 - Cloud & Cloud-Native Computing	3. Technology
04 - Security & Privacy	4. Security
05 - Data Science & Engineering	2. Data Preparation & Utilization
06 - AI, Machine Learning & LLM	2. Data Preparation & Utilization
07 - Fundamental Developer Skills	3. Technology
08 - OS & Network Basics	3. Technology
09 - Programming Concepts & Paradigms	3. Technology
10 - Advanced Programming	3. Technology
11 - Specialized Development Domains	3. Technology
12 - Personal Skills	5. Personal Skill

Principles

This site is built on the following principles:

AI-Driven Development & Human Interaction: As AI agents take over increasingly detailed tasks in software development, the ability to articulate requirements precisely becomes critical. This demands a broad, cross-domain knowledge base — the wider one's understanding of technologies and concepts, the more effectively one can direct AI to produce the right outcomes. At the same time, Digital Transformation involves significant organizational and individual change, so elements from human sciences such as sociology and psychology are incorporated. Understanding human behavior is crucial for driving transformation.

Prioritizing Openness: Open Source Software (OSS) and open formats are favored over proprietary alternatives. This minimizes restrictions and vendor lock-in, which can impede agile decision-making. Furthermore, the availability of source code significantly aids in troubleshooting. Cloud services are included where essential.

Language Agnosticism: Programming languages are treated primarily as tools. Since modern developers can easily work with multiple languages, multi-language workflows have become routine. The focus is on selecting the most suitable language for a specific problem domain or cultural context.

Primary References

These resources serve as the primary references for the content on this site.

- [Level Up Coding](#) - A Medium publication covering software engineering, DevOps, and cloud topics
- [ITNEXT](#) - A Medium publication focused on IT, web development, and DevOps practitioners
- [FAUN](#) - A Medium publication dedicated to cloud native, DevOps, and developer content
- [Thoughtworks Technology Radar](#) - A curated guide to techniques, tools, platforms, and languages worth adopting, trialing, or avoiding
- [Developer Roadmaps](#) - Community-driven roadmaps and learning paths for various engineering roles
- [Golang Weekly](#) - A weekly newsletter covering Go language news, articles, and projects
- [Ruby Weekly](#) - A weekly newsletter covering Ruby language news, articles, and projects

- [Postgres Weekly](#) - A weekly newsletter covering PostgreSQL news, articles, and tools
- [Tony Lixu](#) on Medium - Articles on cloud infrastructure, Kubernetes, and DevOps engineering

01 - Business Transformation

Project Management

Relevant DSS-P Skills

- 1. Business Transformation > 1.3 Management of Transformation Activities > Program / Project Management
- 1. Business Transformation > 1.2 Product Management > Product Scope and Priority Management

Project Planning & Estimation

- [Project management](#) - The process of leading the work of a team to achieve all project goals within the given constraints
 - [PMBOK \(Project Management Body of Knowledge\)](#) - A set of standard terminology and guidelines for project management
 - [Project charter](#) - A statement of the scope, objectives, and participants in a project
 - [Critical chain project management](#) - A method of planning and managing projects that emphasizes the resources (people, equipment, physical space) required to execute project tasks
 - [Program evaluation and review technique \(PERT\)](#) - A statistical tool used in project management to analyze and represent the tasks involved in completing a project
 - [Work breakdown structure](#) - A deliverable-oriented breakdown of a project into smaller components
 - [RACI matrix](#) - A responsibility assignment matrix that describes the participation by various roles in completing tasks or deliverables for a project or business process
 - Responsible, Accountable, Consulted, Informed
 - [Software development effort estimation](#) - The process of predicting the most realistic amount of effort (expressed in terms of person-hours or money)

required to develop or maintain software based on incomplete, uncertain and noisy input

- [Hofstadter's law](#) - A self-referential adage about the difficulty of accurately estimating the time required to complete complex tasks
- [Three-point estimation](#) - A technique used in project management to estimate the likely duration or cost of an activity
- [Planning poker](#) - A consensus-based, gamified technique for estimating, mostly used for timeboxing in Agile principles
- [Fermi problem](#) - An estimation problem in physics or engineering education, designed to teach dimensional analysis or approximation of extreme scientific calculations
- [Systems development life cycle \(SDLC\)](#) - A conceptual model used in project management that describes the stages involved in an information system development project

Prioritization

- [Prioritization](#) - The activity that arranges items or activities in order of urgency
 - [RICE](#) - A simple scoring system for product prioritization that stands for Reach, Impact, Confidence, and Effort
 - [Kano model](#) - A theory for product development and customer satisfaction developed in the 1980s by Noriaki Kano
 - [MoSCoW method](#) - A prioritization technique used in management, business analysis, project management, and software development

Work & Issue Tracking

- Issue Tracking & Task Boards
 - [Jira](#) - A software application used for issue tracking and project management that helps teams plan, assign, track, report, and manage work
 - [JiraCLI](#) - An interactive command line tool for Atlassian Jira that will help you avoid Jira UI to some extent
 - [Linear](#) - A purposeful tool for product development, featuring issues, cycles, and product roadmaps
 - [Fizzy](#) - A modern spin on kanban for tracking anything such as bugs, issues, ideas, and small projects
 - [GitLab Issue Board](#) - A user interface that displays issues in columns that

correspond to their workflow statuses

- [GitLab Service Desk](#) - A feature that enables you to connect with users through email, without requiring them to have a GitLab account
- [Azure Boards](#) - A service that provides a customizable platform for managing work items, allowing teams to collaborate effectively and streamline their workflow
- [GitHub Issues](#) - A tracking tool that helps you manage your work on GitHub
- [Redmine](#) - A free and open source, web-based project management and issue tracking tool
- [OpenProject](#) - The open source project management software for classic, agile or hybrid project management
- Collaborative Work Management
 - [Asana](#) - The platform for human and AI collaboration that helps teams coordinate work and keep projects moving
 - [monday.com](#) - A secure work platform where people and agents drive results together across departments and use cases
 - [ClickUp](#) - An all-in-one productivity platform that brings apps, AI, projects, and chat together to replace all software
 - [Trello](#) - A visual collaboration tool that creates a shared perspective on any project using boards, lists, and cards to organize tasks
 - [Airtable](#) - A platform that combines the flexibility of a spreadsheet with the power of a database to help teams manage their work

Team Collaboration

- Messaging & Meetings
 - [Slack](#) - A cloud-based team collaboration platform that brings conversations, tools, and files together in one place
 - [Microsoft Teams](#) - A collaboration platform that combines workplace chat, meetings, file storage, and application integration
 - [Discord](#) - A voice, video, and text communication service used by communities, friend groups, and businesses to stay connected
 - [Mattermost](#) - An open source collaboration platform for developers, offering secure messaging, project management, and workflow orchestration
 - [Zoom](#) - A video communications platform that provides video meetings, voice calls, webinars, and chat

Process Maturity & Quality Standards

- [CMMI \(Capability Maturity Model Integration\)](#) - A process level improvement training and appraisal program
- [ISO 9001 \(Quality management systems\)](#) - The world's most established quality framework, defining the requirements for a quality management system that organizations wishing to meet the standard must fulfill
- [ISO/IEC 12207 \(Software life cycle processes\)](#) - The international standard for software lifecycle processes that defines all the processes required for developing and maintaining software systems
- [ISO/IEC 15288 \(System life cycle processes\)](#) - The technical standard in systems engineering which covers processes and lifecycle stages
- [ISO/IEC 15504 \(Process assessment\)](#) - A set of technical standards documents for the computer software development process and related business management functions

Product Management

Relevant DSS-P Skills

- 1. Business Transformation > 1.2 Product Management > Requirement Analysis and Management
- 1. Business Transformation > 1.2 Product Management > Product Vision / Roadmap Formulation
- 1. Business Transformation > 1.2 Product Management > Marketing
- 1. Business Transformation > 1.3 Management of Transformation Activities > Product Lifecycle Management

Product Strategy & Roadmapping

- [Product management](#) - The business process of planning, developing, launching, and managing a product or service
- [Lean startup](#) - A methodology for developing businesses and products that aims to shorten product development cycles and rapidly discover if a proposed business model is viable
- [Jobs to Be Done \(JTBD\)](#) - A lens that reveals the circumstances that drive people and organizations toward and away from decisions
- [Crowdfunding](#) - The practice of funding a project or venture by raising money from a large number of people, typically via the internet

- **Business model** - The rationale of how an organization creates, delivers, and captures value, in economic, social, cultural or other contexts
 - **Direct-to-consumer** - A business model of selling products directly to customers and thereby bypassing any third-party retailers, wholesalers, or intermediaries
 - **Subscription business model** - A business model in which a customer must pay a recurring price at regular intervals for access to a product or service
 - **Business model canvas** - A strategic management template for developing new or documenting existing business models
 - **Lean Canvas** - A one-page business modeling tool for entrepreneurs to quickly outline their business idea
- **Technology roadmap** - A flexible planning schedule to support strategic and long-range planning, by matching short-term and long-term goals with specific technology solutions
 - **Aha!** - A suite of product development software that helps teams build and market products customers love

Requirements Analysis

- **Business analysis** - A professional discipline focused on identifying business needs and determining solutions to business problems
- **Requirements analysis** - The process of determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders
 - **Requirement** - A documented need of what a product or service should be or do
 - **Non-functional requirement** - A requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors
- Related Standards
 - **ISO/IEC 25010 (Systems and software Quality Requirements and Evaluation)** - The international standard for systems and software quality requirements and evaluation that defines system and software quality models

Marketing & Customer Experience

- **Marketing** - A process of acquiring, satisfying, and retaining customers, often involving creating, communicating, delivering, and exchanging offerings that have value for various stakeholders
 - **Market research** - An organized effort to gather information about target markets and customers

- **SEO** - The process of improving the quality and quantity of website traffic to a website or a web page from search engines
 - **Google Search Central** - The home for everything you need to help users find your site on Google Search
- **Marketing mix** - A foundation model for businesses, historically centered around product, price, place, and promotion
- **Fear of missing out (FOMO)** - The feeling of apprehension that one is either not in the know about or missing out on information, events, experiences, or life decisions that could make one's life better
- **Fear, uncertainty, and doubt (FUD)** - A manipulative propaganda tactic used in sales, marketing, public relations, politics, polling, and cults
- **Mere-exposure effect** - A psychological phenomenon by which people tend to develop a liking or disliking for things merely because they are familiar with them
- Advertising
 - Indicators
 - **Click through rate** - The ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement
 - **Conversion rate** - The percentage of users who take a desired action
- Concepts and Frameworks
 - **Brand** - A name, term, design, symbol or any other feature that distinguishes one seller's goods or service from those of other sellers
 - **Customer experience** - A customer's cognitive, emotional, sensory, and behavioral responses during all stages of interaction with a product or service
 - **Customer service** - The assistance and advice provided by a company to those who buy or use its products or services, either in person or remotely
 - **Value chain** - A progression of activities that a business or firm performs in order to deliver goods and services of value to an end customer
- Tools for Strategy
 - **Value proposition canvas** - A tool to help you create products and services customers want

Marketing Technology

- Tag Management

- [Google Tag Manager](#) - A tag management system that allows you to quickly and easily update measurement codes and related code fragments known as tags on your website or mobile app
- Web Analytics
 - [Google Analytics](#) - The go-to platform for millions of website and app owners seeking to gain a deeper understanding of their website and app performance
 - [Plausible](#) - Intuitive, lightweight and open source web analytics
 - [Umami](#) - A simple, fast, privacy-focused alternative to Google Analytics
 - [Ackee](#) - Self-hosted, privacy-focused analytics tool for those who care about privacy
- Customer Data Platforms
 - [Customer data platform](#) - A software system that aggregates and organizes customer data from various touchpoints to build a unified customer profile
 - [Twilio Segment](#) - A customer data platform that enables businesses to collect, unify, and activate customer data across multiple sources and destinations to create personalized experiences
- Marketing Automation
 - [Marketing automation](#) - Software platforms and technologies designed for marketing departments and organizations to automate repetitive tasks
 - [Klaviyo](#) - An American marketing automation platform and email marketing service
- Advertising Platforms
 - [Google Ads](#) - An online advertising platform where advertisers bid to display brief advertisements, service offerings, product listings, or videos to web users
 - [Google AdSense](#) - A program run by Google through which website publishers in the Google Network of content sites serve text, images, video, or interactive media advertisements that are targeted to the site content and audience
- Experimentation & Optimization
 - [Optimizely](#) - A leading digital experience platform (DXP) that provides a single, unified platform that offers you the scalability and security you need to drive your business into the future
- Email Marketing & Delivery
 - [SendGrid](#) - A cloud-based email delivery platform that provides reliable transactional and marketing email delivery at scale
 - [Mailchimp](#) - An all-in-one marketing platform for small businesses that helps

manage and talk to clients, customers, and audiences with email marketing

- [listmonk](#) - Self-hosted newsletter and mailing list manager
- [BillionMail](#) - An open-source MailServer and email marketing solution that is fully self-hosted and dev-friendly

Metrics & Performance

Relevant DSS-P Skills

- 1. Business Transformation > 1.2 Product Management > Design and Operation of Product Performance Metrics
- 1. Business Transformation > 1.1 Strategy Understanding and Architecture Design > Business Value Definition / ROI Estimation and Decision Support

Goal Setting Frameworks

- [Goal setting](#) - The process of developing an action plan designed to motivate and guide a person or group toward a goal
 - [SMART goals](#) - A mnemonic acronym, used to guide in the setting of objectives or goals, for example in project management, employee-performance management and personal development
 - Specific: Targeting a particular area for improvement
 - Measurable: Quantifying, or at least suggesting, an indicator of progress
 - Assignable: Defining responsibility clearly
 - Realistic: Outlining attainable results with available resources
 - Time-related: Including a timeline for expected results
 - [FAST goals](#) - A framework for goals that are Frequently discussed, Ambitious in scope, Specific in metrics, and Transparent for everyone to see
 - [GROW model](#) - A simple method for goal setting and problem solving
 - [OKRs](#) - A goal-setting framework used by individuals, teams, and organizations to define measurable goals and track their outcomes
 - [KPIs](#) - A type of performance measurement used to evaluate the success of an organization or of a particular activity (such as projects, programs, products and other initiatives) in which it engages
 - [Goodhart's law](#) - An adage often stated as, "When a measure becomes a target, it ceases to be a good measure"

Product & Customer Metrics

- [Net Promoter Score](#) - A market research metric that is based on a single survey question asking respondents to rate the likelihood that they would recommend a company, product, or a service to a friend or colleague
- [Rubric](#) - A scoring tool used to evaluate the quality of responses

Engineering Performance Metrics

- [SPACE framework](#) - A framework that provides a way to think about developer productivity in a more holistic way, encompassing Satisfaction and well-being, Performance, Activity, Communication and collaboration, and Efficiency and flow
- [The Four Keys of DORA](#) - A set of metrics used to measure DevOps performance, consisting of Deployment Frequency, Lead Time for Changes, Change Failure Rate, and Time to Restore Service

Enterprise Strategy & Architecture

Relevant DSS-P Skills

- 1. Business Transformation > 1.1 Strategy Understanding and Architecture Design > Understanding of Business Environment and Management Strategy
- 1. Business Transformation > 1.1 Strategy Understanding and Architecture Design > Business and Enterprise Architecture Design
- 1. Business Transformation > 1.3 Management of Transformation Activities > Architecture Management & Governance
- 1. Business Transformation > 1.3 Management of Transformation Activities > Change Management
- 1. Business Transformation > 1.3 Management of Transformation Activities > Risk & Compliance

Enterprise Architecture

- [Enterprise architecture](#) - A well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a comprehensive approach at all times, for the successful development and execution of strategy
 - [TOGAF standard](#) - A proven Enterprise Architecture methodology and framework used by the world's leading organizations to improve business efficiency

- [Zachman Framework](#) - An ontology – a theory of the existence of a structured set of essential components of an object
- [ArchiMate](#) - An open and independent modelling language for Enterprise Architecture that is supported by different tool vendors and consulting firms
 - [Archi](#) - A free, open source, cross-platform tool and editor to create ArchiMate models
- [Enterprise modeling](#) - The process of building models of whole or part of an enterprise with process models, data models, resource models and or new ontologies
 - [BPMN](#) - A graphical notation for specifying business processes in a Business Process Diagram, providing a standard comprehensible to business users yet representing complex process semantics for technical users
 - [SysML](#) - A general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities
 - [Eclipse Capella](#) - A powerful and extensible MBSE software tool that leverages a field-proven language and method to successfully design the architecture of complex systems

Enterprise Applications

- [Enterprise resource planning](#) - The integrated management of main business processes, often in real time and mediated by software and technology
 - [SAP ERP](#) - A comprehensive software system that streamlines processes, improves productivity, and provides real-time insights across your entire organization
 - [Odoo](#) - A suite of open source business applications covering areas such as CRM, ERP, accounting, and more
 - [ERPNext](#) - A 100% open-source ERP with a modern, comprehensive, and user-friendly enterprise resource planning solution
- [Customer relationship management](#) - A strategic process that organizations use to manage, analyze, and improve their interactions with customers
 - [EspoCRM](#) - An open-source web application for managing and evaluating all company relationships
 - [HubSpot](#) - A customer platform that helps businesses grow by connecting their marketing, sales, and service tools to a shared database
 - [Salesforce](#) - A customer relationship management solution that brings

companies and customers together, providing one integrated CRM platform for all departments

- [Zendesk](#) - A customer service software and support ticket system that helps businesses build better customer relationships through multi-channel support
- [Atlas](#) - A bespoke AI for customer support that delivers fast, accurate, and measurable support tailored to tools and workflows
- [SuiteCRM](#) - A free and open-source Customer Relationship Management (CRM) software solution providing a 360-degree view of customers and business
- [Supply chain management](#) - The management of the flow of goods and services, between businesses and locations, including the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption
- [Human resource management](#) - The strategic and coherent approach to the effective and efficient management of people in a company or organization such that they help their business gain a competitive advantage
 - [Competence](#) - The set of demonstrable characteristics and skills that enable and improve the efficiency or performance of a job
- [Contract management](#) - The process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk
- [E-commerce](#) - The activity of electronically buying or selling products on online services or over the Internet
 - [Shopify](#) - A Canadian multinational e-commerce company that provides a proprietary e-commerce platform for online stores and retail point-of-sale systems
 - [Stripe](#) - A financial infrastructure platform for businesses that provides payment processing software and APIs for e-commerce websites and mobile applications

Organizational Change Management

- [Organizational structure](#) - A system that outlines how certain activities are directed in order to achieve the goals of an organization
- [Diffusion of innovations](#) - A theory that seeks to explain how, why, and at what rate new ideas and technology spread
- [Kotter's 8-step change model](#) - A set of tools and strategies designed to help organizations effectively implement and sustain change
- [Prosci ADKAR Model](#) - A goal-oriented change management model that guides

individual and organizational change

Strategic Analysis & Business Intelligence

- Strategic Analysis Frameworks
 - [MECE principle](#) - The grouping principle for separating a set of items into subsets that are mutually exclusive (ME) and collectively exhaustive (CE)
 - [SWOT analysis](#) - A decision-making technique used in strategic planning and management that identifies the strengths, weaknesses, opportunities, and threats of an organization or project
 - [PEST analysis](#) - A framework of external macro-environmental factors (political, economic, social and technological) used in strategic management and market research
 - [Porter's five forces analysis](#) - A method of analyzing the competitive environment of a business, rooted in industrial organization economics, that identifies five forces determining competitive intensity and industry attractiveness
- Business Intelligence Platforms
 - [Tableau](#) - The visual analytics platform that helps people see, understand, and act on data to solve problems
 - [Metabase](#) - The querying and visualization layer for your database, made to fit startup's production DB to massive data warehouses
 - [Power BI](#) - A unified, scalable platform for self-service and enterprise business intelligence
 - [DAX](#) - A programming language that is used throughout Microsoft Power BI for creating calculated columns, measures, and custom tables
 - [Looker Studio](#) - A business intelligence and data visualization platform, formerly known as Google Data Studio, that turns your data into customizable, informative reports and dashboards
 - [Exploratory Desktop](#) - A Simple and Easy-to-Use UI experience to access various data sources, clean and transform data, visualize and analyze data

Risk Management

- [Risk management](#) - The identification, evaluation, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability or impact of unfortunate events or to maximize the realization of opportunities

- [Business continuity planning](#) - The process an organization undergoes to create a prevention and recovery system from potential threats such as natural disasters or cyber-attacks
 - [IT disaster recovery](#) - The process of resuming normal IT operations after a disruptive event, such as a natural disaster, cyberattack, or equipment failure
 - [ISO 22301 \(Business continuity management systems\)](#) - The international standard for business continuity management systems that specifies requirements to protect against, respond to, and recover from disruptive incidents
- [Project risk management](#) - The process of identifying, analyzing, and then responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal
- [Financial risk management](#) - The practice of protecting economic value in a firm by managing exposure to financial risk - principally credit risk and market risk, as well as some aspects of operational risk
- [ISO 31000 \(Risk management\)](#) - A set of international standards for risk management that provides a consistent vocabulary and methodology for assessing and managing risk

IT Service Management

- [IT service management](#) - The activities that are performed by an organization to design, build, deliver, operate and control information technology (IT) services offered to customers
 - [ITIL \(Information Technology Infrastructure Library\)](#) - A set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business
 - [ISO/IEC 20000 \(Service management\)](#) - The international standard for IT service management that specifies the requirements for establishing, implementing, maintaining, and continually improving a service management system
 - [ServiceNow](#) - A cloud-based, AI-powered platform for digital workflows that connects people, functions, and systems across the enterprise

Knowledge & Content Management

Relevant DSS-P Skills

- 1. Business Transformation > 1.3 Management of Transformation Activities >

Business Modeling and Collaboration

Personal Knowledge Management

- [Zettelkasten](#) - A system of note-taking and personal knowledge management for research, study, and writing, consisting of small, interconnected items of information stored on slips or cards
- Note-Taking Tools
 - [Obsidian](#) - A free and flexible application for private thoughts, allowing users to store notes on their device, connect ideas, and organize knowledge with a customizable interface through plugins and themes
 - [Memos](#) - An open-source, self-hosted note-taking tool built for quick capture that is markdown-native and lightweight
- Feed Aggregation
 - [FreshRSS](#) - A free, self-hostable aggregator for RSS and Atom feeds

Collaborative Workspaces & Wikis

- [Wiki software](#) - A collaborative software that runs a wiki, which allows users to create and collaboratively edit pages or entries via a web browser
 - [MediaWiki](#) - A free and open-source wiki software
 - [Ibis](#) - A federated encyclopedia which uses the ActivityPub protocol, just like Mastodon or Lemmy
 - [LeafWiki](#) - A real wiki app without the heavy stack, storing Markdown files on disk without requiring a database server
- Team Workspaces
 - [Notion](#) - An all-in-one workspace for notes, docs, wikis, projects, and collaboration that combines knowledge management with task and project tracking
 - [Coda](#) - An all-in-one collaborative workspace that brings teams and tools together for a more organized work day
 - [Confluence](#) - One place for all your ideas, docs, knowledge, and teammates
 - [Outline](#) - The knowledge base platform that helps teams organize documents, collaborate in real-time, and search across their workspace with AI-powered question answering

Content Management Systems

- [Content management system](#) - A computer software used to manage the creation and modification of digital content
 - [WordPress](#) - A free and open-source content management system written in hypertext preprocessor language and paired with a MySQL or MariaDB database with supported HTTPS
 - [Drupal](#) - A free and open-source web content management system written in PHP and distributed under the GNU General Public License

Content Collaboration & File Sync

- [SharePoint](#) - A web-based collaboration and document management platform that enables organizations to securely store, share, and manage content
- [Nextcloud](#) - The industry-leading, fully open-source, on-premise content collaboration platform
- [Box](#) - An enterprise cloud content management platform that enables organizations to securely manage and share content while collaborating with internal and external users
- [Dropbox](#) - A file hosting service that offers cloud storage, file synchronization, personal cloud, and client software

Enterprise AI & Productivity

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Understanding and Utilization of Data and AI
- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Data / AI Utilization Strategy Design

Autonomous AI Workers

- [Claude Cowork](#) - An agentic AI system for knowledge work that autonomously plans and executes multi-step tasks across files, documents, and web applications directly on your computer
- [claw-empire](#) - A local-first AI agent office simulator that orchestrates CLI, OAuth, and API-connected agents as a virtual autonomous company
- [Sistava](#) - An AI employee platform for managing teams of AI workers collaborating

through structured sprints, OKRs, and KPIs with persistent memory and real tool access

Enterprise AI Assistants

- [Amazon Q Business](#) - A generative AI-powered assistant for enterprises to find information, gain insights, and take action at work, integrating with company data and applications
- [Microsoft 365 Copilot](#) - An AI assistant for work that supercharges productivity and creativity, reengineers business processes, and empowers you to securely transform your business into an AI-powered organization
- [Notion AI](#) - An integrated AI assistant for workspaces that provides writing assistance, workspace Q&A, and autonomous agents for task automation
- [Gemini for Google Workspace](#) - An AI-powered assistant for Docs, Gmail, Sheets, and Slides that helps you write, visualize, and organize your work across the platform
- [Claude for Enterprise](#) - A secure and scalable way for organizations to use AI with administrative controls, single sign-on (SSO), and role-based access to Claude's latest models
- [ChatGPT Enterprise](#) - An enterprise-grade AI assistant with unlimited higher-speed access to GPT-4, longer context windows, and advanced data analysis capabilities
- [Glean](#) - An AI-powered search and assistant for the enterprise that connects to all of your company's apps and data to find exactly what you need

Self-Hosted AI Platforms

- [AnythingLLM](#) - The all-in-one AI application for everyone
- [Dify](#) - An open-source LLM app development platform
- [Flowise](#) - An open source generative AI development platform for building AI agents, LLM orchestration, and more
- [LibreChat](#) - The open-source AI platform that brings together all your AI conversations in one unified, customizable interface
- [OpenWebUI](#) - An extensible, feature-rich, and user-friendly self-hosted AI platform designed to operate entirely offline

AI Content & Document Generation

- [Beautiful.ai](#) - The AI presentation platform built for enterprise teams that keeps every deck on-brand and ready to share across the whole company

- [Gamma](#) - A new medium for presenting ideas, powered by AI
- [Napkin AI](#) - The visual AI for business storytelling that turns your text into visuals so sharing your ideas is quick and effective
- [NotebookLM](#) - The AI research tool and thinking partner that can analyze your sources, turn complexity into clarity and transform your content

AI Agent Registries

- Agent Skills Registries
 - [The Agent Skills Directory](#) - An open agent skills ecosystem providing reusable capabilities for AI agents
 - [Anthropic Agent Skills](#) - A public repository containing Anthropic's implementation of skills for Claude, including instructions, scripts, and resources that enable specialized tasks and repeatable workflows
 - [SkillsMP \(Skills Management Platform\)](#) - A community-driven marketplace designed for discovering and sharing modular AI agent capabilities based on the open SKILL.md standard
- MCP (Model Context Protocol) Registries
 - [Official MCP Registry](#) - A collection of official reference MCP server implementations maintained by the Model Context Protocol organization
 - [MCP Registry](#) - A searchable web directory of published MCP servers
 - [Claude Plugins](#) - A directory that bundles tools, skills, and integrations for one-click installation

Human-Centered Design

Relevant DSS-P Skills

- 1. Business Transformation > 1.4 Design > Customer / User / Stakeholder Understanding
- 1. Business Transformation > 1.4 Design > Digital Product Design

Core Principles & User Experience (UX)

- [User experience](#) - A person's emotions and attitudes about using a particular product, system or service
- [Usability](#) - The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context

of use

- **User interface design** - A craft in which designers perform an important function in creating the user experience
- **Design thinking** - The set of cognitive, strategic and practical processes by which design concepts are developed
- **Accessibility** - The design of products, devices, services, or environments for people with disabilities
 - **Accessibility Object Model (AOM)** - A JavaScript API to allow developers to modify (and eventually explore) the accessibility tree for an HTML page
 - **WAI-ARIA** - The Accessible Rich Internet Applications suite of web standards
- Prototyping
 - **Paper prototyping** - A widely used method in the user-centered design process, a process that helps developers to create software that meets the user's expectations and needs
 - **Website wireframe** - A skeletal outline of a webpage

User Research & Ideation

- Research Methodologies
 - **Extreme users** - A user-centered design methodology that focuses on studying users at the extremes of the usage spectrum to inform design solutions that benefit both edge cases and typical users
 - **Card sorting** - A method used to help design or evaluate the information architecture of a site
 - **A/B testing** - A way to compare multiple versions of a single variable, for example by testing a subject's response to variant A against variant B, and determining which of the variants is more effective
 - **Diary studies** - A research method in which people record their experiences and activities over time
 - **Persona** - A fictional character created to represent a user type relationship
- Ideation Techniques
 - **Affinity diagram** - A business tool used to organize ideas and data
 - **Brainstorming** - A group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members
 - **SCAMPER** - A structured way of assisting students to think out of the box and

enhance their knowledge

- [How Might We \(HMW\)](#) - A design thinking method that allows designers to reframe and open up their problem statements for efficient, targeted and innovative ideation sessions to help solve design challenges

Cognitive & Behavioral Psychology

- Psychological Models
 - [Seven stages of action](#) - An idealized description of the cognitive and physical steps an individual takes to achieve a goal
 - 1: Forming the target
 - 2: Forming the intention
 - 3: Specifying an action
 - 4: Executing the action
 - 5: Perceiving the state of the world
 - 6: Interpreting the state of the world
 - 7: Evaluating the outcome
- Cognitive Processes
 - [Attention](#) - The cognitive process of selectively concentrating on one aspect of the environment while ignoring other things
 - [Metacognition](#) - An awareness of one's thought processes and an understanding of the patterns behind them
- Interaction Principles & Laws
 - [Principle of least astonishment](#) - A general principle that states that the result of performing some operation should be obvious, consistent, and predictable, based upon the name of the operation and other context
 - [Affordance](#) - A property of an object that indicates how it can be used
 - [Stroop effect](#) - A demonstration of interference in the reaction time of a task
 - [Fitts's law](#) - A predictive model of human movement primarily used in human-computer interaction and ergonomics

Visual Design & Typography

- Typography
 - [Typography](#) - The art and technique of arranging type to make written language legible, readable and appealing when displayed

- [Web Typography](#) - The use of fonts on the World Wide Web
- [Microsoft Typography](#) - A comprehensive resource for font technology and typefaces, providing technical specifications, developer tools, and design guidelines for Microsoft products
- Visual Foundations
 - [Color space](#) - A specific organization of colors
 - [ICC profile](#) - A set of data that characterizes a color input or output device, or a color space
 - [sRGB](#) - A standard RGB color space that HP and Microsoft created cooperatively in 1996 for use on monitors, printers, and the Internet
 - [HSL and HSV](#) - The two most common cylindrical-coordinate representations of points in an RGB color model
 - [Lucide](#) - A beautiful and consistent icon library for various platforms and frameworks
- Font Rendering & Technologies
 - Font Standards
 - [TrueType](#) - An outline font standard developed by Apple and Microsoft in the late 1980s as a competitor to Adobe's Type 1 fonts used in PostScript
 - [OpenType](#) - A scalable computer font format developed by Microsoft and Adobe as an extension of the TrueType format, supporting advanced typographic features and multi-platform compatibility
 - [WOFF \(Web Open Font Format\)](#) - A font format for use in web pages, developed by Mozilla and others, that provides a compressed wrapper for TrueType and OpenType fonts to improve web performance
 - [Variable Fonts](#) - An evolution of the OpenType font specification that enables a single font file to behave like multiple fonts by defining variations in weight, width, and other axes
 - Open Fonts
 - [Noto Fonts](#) - A global font collection for all modern and ancient languages
 - [Orbitron](#) - A geometric sans-serif typeface intended for display purposes
 - Libraries & Engines
 - [FreeType](#) - A freely available software library to render fonts
 - [HarfBuzz](#) - A widely used open-source text-shaping engine that converts Unicode text into the glyphs and positions required for proper rendering across various scripts and languages

- [Pango](#) - An open-source library for laying out and rendering of text, with an emphasis on internationalization and support for complex scripts
- [Fontconfig](#) - A library for configuring and customizing font access, used primarily on Linux and other Unix-like systems to provide consistent font matching and substitution
- Rendering Technologies & APIs
 - [ClearType](#) - A subpixel rendering technology developed by Microsoft to improve the readability of text on liquid-crystal displays (LCDs) by utilizing the individual subpixels of each pixel
 - [DirectWrite](#) - A high-performance text-layout and font-rendering API from Microsoft that supports hardware-accelerated rendering and high-quality typography for modern applications

Design Systems & Design Tools

- Visual Design Tools
 - [Claude Design](#) - A visual design tool that lets users collaborate with Claude to create polished designs, prototypes, slides, and marketing materials through natural conversation and iterative refinement
 - [Figma Design](#) - A powerful, collaborative design tool for teams
 - [Locofy.ai](#) - Design to code in a flash
 - [Uizard](#) - An AI-powered design tool that lets you generate prototypes, screens, and themes from text prompts
 - [Anything](#) - An AI app builder that turns your words into mobile apps, sites, tools, and products built with code
 - [v0](#) - An AI-powered full-stack web application builder that generates working applications in minutes and publishes them as live websites in seconds
- Design Systems & Guidelines
 - [Material Design](#) - Google's open-source design system for building beautiful, usable products
 - [Apple HIG](#) - A set of recommendations to help you create apps that look and behave consistently across all Apple platforms
 - [GNOME HIG](#) - A guide for creating high-quality, consistent, and usable applications for the GNOME desktop

Web Experience & Performance

- [Responsive web design](#) - An approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes
- [Core Web Vitals](#) - The subset of Web Vitals that apply to all web pages, should be measured by all site owners, and will be surfaced across all Google tools
 - Largest Contentful Paint (LCP)
 - Interaction to Next Paint (INP)
 - Cumulative Layout Shift (CLS)

Economics, Game Theory & Finance

Relevant DSS-P Skills

- 1. Business Transformation > 1.1 Strategy Understanding and Architecture Design > Understanding of Business Environment and Management Strategy
- 1. Business Transformation > 1.1 Strategy Understanding and Architecture Design > Business Value Definition / ROI Estimation and Decision Support

Economics & Game Theory

- [Market](#) - A composition of systems, institutions, procedures, social relations or infrastructures whereby parties engage in exchange
- [Inflation](#) - An increase in the general price level of goods and services in an economy over a period of time
- [Prospect theory](#) - A theory of behavioral economics and behavioral finance which states that people make decisions based on the potential value of losses and gains rather than the final outcome
- [Sunk cost](#) - A cost that has already been incurred and cannot be recovered
- [Choice architecture](#) - The design of different ways in which choices can be presented to decision makers, and the impact of that presentation on decision-making
 - [Nudge theory](#) - A concept in behavioral economics and related behavioral sciences that proposes adaptive designs of the decision environment as ways to influence the behavior and decision-making of groups or individuals
- [Principal-agent problem](#) - The conflict in priorities between a person or group and the representative authorized to act on their behalf
- [Information asymmetry](#) - A situation in which one party in a transaction has more

or better information than the other

- **Induced demand** - The phenomenon that after supply increases, more of a good is consumed
- **Metcalfe's law** - The value of a telecommunications network is proportional to the square of the number of connected users of the system (n^2)
 - **Network effect** - The phenomenon by which the value or utility a user derives from a good or service depends on the number of users of compatible products
- **Braess's paradox** - The observation that adding one or more roads to a road network can slow down overall traffic flow through it
- **Nash equilibrium** - A solution concept of a non-cooperative game involving two or more players in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only their own strategy
- **Pareto efficiency** - A state of allocation of resources from which it is impossible to reallocate so as to make any one individual or preference criterion better off without making at least one individual or preference criterion worse off
- **Operations research** - A discipline that deals with the development and application of analytical methods to improve management and decision-making

Corporate Finance & Markets

- **Currency** - A standardization of money in any form, in use or circulation as a medium of exchange
- **Interest** - The payment from a debtor or deposit-taking financial institution to a lender or depositor of an amount above repayment of the principal sum (that is, the amount borrowed), at a particular rate
- **Central bank** - An institution that manages the monetary policy of a country or monetary union
- **Revenue model** - A framework for generating financial income
- **Financial capital** - An economic resource measured in terms of money used by entrepreneurs and businesses to buy what they need to make their products or to provide their services
 - **Venture capital** - A form of private equity financing that is provided by venture capital firms or funds to startups, early-stage, and emerging companies that have been deemed to have high growth potential
- **Markets & Securities**
 - **Stock market** - The aggregation of buyers and sellers of stocks, which represent

ownership claims on businesses

- **Stock** - Shares that divide ownership of a corporation, representing fractional ownership and typically conferring rights to earnings, liquidation proceeds, or voting power
- **Dividend** - The distribution of profits by a corporation to its shareholders from current year profit or retained earnings
- **Contracts**
 - **Credit** - The trust which allows one party to provide money or resources to another party wherein the second party does not reimburse the first party immediately
 - **Debt** - An obligation that requires one party, the debtor, to pay money or otherwise return value to another party, the creditor
 - **Discounting** - A mechanism in which a debtor obtains the right to delay payments to a creditor, for a defined period of time, in exchange for a charge or fee
 - **Bond** - A type of security under which the issuer (debtor) owes the holder (creditor) a debt, and is obliged – depending on the terms – to repay the principal of the bond at the maturity date and pay interest over a specified time
 - **Spot** - A contract of buying or selling a commodity, security or currency for immediate settlement
 - **Futures** - A standardized legal contract to buy or sell something at a predetermined price for delivery at a specified time in the future
 - **Option** - A contract which conveys to its owner, the holder, the right, but not the obligation, to buy or sell a specific quantity of an underlying asset or instrument at a specified strike price on or before a specified date
- **Cryptocurrency** - A type of currency which uses digital files as money

Accounting & Financial Reporting

- **Accounting Fundamentals**
 - **Asset** - A resource owned or controlled by a business or economic entity that can be used to produce positive economic value
 - **Liability** - A quantity of value that a financial entity owes and is expected to deliver in the future to satisfy a present obligation arising from past events
 - **Equity** - An ownership interest in property that may be subject to debts or other liabilities, measured by subtracting liabilities from the value of assets owned

- **Revenue** - The total amount of income generated by the sale of goods and services related to the primary operations of a business
- **Depreciation** - The decrease in the value of assets and the method used to reallocate the cost of a tangible asset over its useful life span
- **Accrual** - An accounting method that recognizes revenues and expenses when they are earned or incurred, not necessarily when cash is received or paid
- Financial Statements & Metrics
 - **Balance sheet** - A summary of the financial balances of an individual or organization
 - **Income statement** - One of the financial statements of a company and shows the company's financial performance for a specific period of time
 - **Cash flow statement** - A financial statement that shows how changes in balance sheet accounts and income affect cash and cash equivalents
 - **Return on investment** - The ratio between net income (over a period) and investment (costs resulting from an investment of some resources at a point in time)
 - **Net present value** - A way of measuring the value of an asset that has cashflow by adding up the present value of all the future cash flows that asset will generate
 - **EBITDA** - A measure of a company's profitability of the operating business only, before any effects of indebtedness, state-mandated payments, and costs required to maintain its asset base
 - **Operating margin** - The ratio of operating income to net sales, usually expressed in percent
 - **Burn rate** - The rate at which a company consumes its cash, typically expressed monthly and used for startups to measure how fast a company will use up its shareholder capital
 - **Liquidity** - A market's feature whereby an individual or firm can quickly purchase or sell an asset without causing a drastic change in the asset's price
 - **Valuation** - The process of determining the value of a potential investment, asset, or security
- Accounting Standards & Processes
 - **Generally Accepted Accounting Principles** - Accounting standards that prescribe in detail what accruals must be made, how financial statements are to be presented, and what additional disclosures are required
 - **Audit** - An independent examination of financial information of any entity

conducted with a view to express an opinion thereon

02 - Web Application Development

Web Platform Fundamentals

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Web Application Core Technology

Web Concepts

- **World Wide Web** - An information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet
 - **Hypertext** - A text displayed on a computer display or other electronic devices with references (hyperlinks) to other text that the reader can immediately access
 - **Semantic Web** - An extension of the World Wide Web that allows Internet data to be machine-readable through standards set by the W3C, enabling automated agents to process information more intelligently
 - **URI** - A unique sequence of characters that identifies a logical or physical resource
 - **URL** - A standard that defines URLs, domains, IP addresses, the application/x-www-form-urlencoded format, and their API
- Core Web Protocols & Languages
 - **HTTP** - An application protocol for distributed, collaborative, hypermedia information systems
 - **HTTP cookie** - A small piece of data that a server sends to a user's web browser
 - **HTML** - The World Wide Web's core markup language
 - **CSS** - A simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents
- Real-time & Messaging Protocols
 - **WebSockets** - A technology that makes it possible to open a two-way interactive communication session between the user's browser and a server
 - **WebRTC** - A free and open-source project providing web browsers and mobile applications with real-time communication (RTC)

- [Server-sent events](#) - A technology to enable servers to push data to web pages over HTTP or using dedicated server-push protocols
- [MQTT](#) - A lightweight, publish-subscribe, machine to machine network protocol for message queue/message queuing service
- [AMQP](#) - An open standard application layer protocol for message-oriented middleware
- Data & Event Specifications
 - [CloudEvents](#) - A specification for describing event data in a common way
 - [JSON Merge Patch](#) - A JSON format that describes changes to be made to a target JSON document
 - [OpenAPI spec](#) - A standard, language-agnostic interface to HTTP APIs
 - [TypeSpec](#) - A minimal language that helps developers describe API shapes in a familiar way
 - API Tooling
 - [Redocly CLI](#) - An open-source command-line tool that helps you lint, bundle, and preview OpenAPI definitions
- Web Performance Concepts
 - [DNS Prefetching](#) - A mechanism to resolve domain names before a user tries to follow a link
- Web Application Types
 - [Progressive web app](#) - A type of application software delivered through the web, built using common web technologies including HTML, CSS, JavaScript, and WebAssembly

Browser Technologies & DOM

- Browsers
 - [Chrome](#) - A freeware, cross-platform web browser developed by Google
 - [Chromium](#) - An open-source browser project that aims to build a safer, faster, and more stable way for all users to experience the web
 - [Firefox](#) - A free and open-source web browser developed by the Mozilla Foundation
 - [w3m](#) - A text-based web browser as well as a pager
 - [EWW](#) - The Emacs Web Wowser, a web browser for Emacs
- Rendering Engines

- [WebKit](#) - The framework for displaying rich, interactive web content in your apps
- [Gecko](#) - The web browser engine developed by Mozilla
- [Blink](#) - The rendering engine used by Chromium
- [Servo](#) - A modern, high-performance browser engine designed for both application and embedded use
- Scripting Engines
 - [V8 \(JavaScript engine\)](#) - Google's open source high-performance JavaScript and WebAssembly engine, written in C++
 - [JavaScriptCore](#) - The JavaScript engine that powers Safari and other apps on Apple platforms
- Client Scripting APIs
 - [XMLHttpRequest \(XHR\)](#) - An API that provides scripted client functionality for transferring data between a client and a server
 - [Fetch Standard](#) - A living standard that defines requests, responses, and the process that binds them: fetching
 - [Canvas API](#) - The means for drawing graphics via JavaScript and the HTML `<canvas>` element
 - [WebGL API](#) - A JavaScript API for rendering high-performance interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins
 - [Web Neural Network API \(WebNN\)](#) - An emerging web standard that allows web apps and frameworks to accelerate deep neural networks with on-device hardware such as GPUs, CPUs, or NPUs
- Site Analyzers
 - [Wappalyzer](#) - A technology profiler that shows you what websites are built with

Web Application Architectures

- [Single-page application](#) - A web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server
- [Multi-page application](#) - A traditional web structure with multiple pages that are independently downloaded from the server, each having its own URL and loaded separately when requested by the user
- [Microfrontend](#) - An architectural pattern for web development where independently developed frontends are composed into a greater whole
- [Islands Architecture](#) - A frontend pattern that renders pages to fast, static HTML with selective "islands" of JavaScript added only where interactivity is needed

- [Backend for Frontend](#) - An architectural pattern where separate backend services are created specifically for different frontend applications
- [Multitier architecture](#) - A client-server architecture where different levels of software architecture are physically separated into presentation, application processing, and data management functions
- [Server-side rendering](#) - An approach where static HTML is sent from the server to the client, and client-side JavaScript then makes the web page dynamic by attaching event handlers in a process called hydration
- [Incremental Static Regeneration](#) - A technique that enables static-generation on a per-page basis without needing to rebuild the entire site, allowing updates to static content after deployment
- [JAMstack](#) - An architectural approach that decouples the web experience layer from data and business logic, improving flexibility, scalability, performance, and maintainability

Frontend Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Front-end System Development
- 1. Business Transformation > 1.4 Design > Digital Product Design

UI Frameworks & Core Libraries

- Core SPA Frameworks
 - [React](#) - The library for web and native user interfaces
 - Core Concepts
 - [Component](#) - A fundamental building block used to create user interfaces
 - [Props](#) - A mechanism for passing data from a parent component to a child component
 - [Children](#) - A special prop that allows components to be composed
 - [Key Props](#) - A special string attribute you need to include when creating lists of elements
 - [Rendering](#) - The process of React asking your components to describe what they want to look like
 - [Event Handler](#) - A function that is executed in response to an event
 - [State](#) - A JavaScript object that stores a component's dynamic data

- **Controlled Component** - A component where React state controls the value of an input element
- **Hooks** - A set of functions that let you "hook into" React state and lifecycle features from function components
- **Strict Mode** - A tool for highlighting potential problems in an application
- **Side-effect** - A term that refers to any operation that affects something outside of the function being executed
- **Refs** - A feature that provides a way to access DOM nodes or React elements created in the render method
- **Context** - A way to pass data through the component tree without having to pass props down manually at every level
- **Portals** - A feature that provides a first-class way to render children into a DOM node that exists outside the DOM hierarchy of the parent component
- **Suspense** - A component that lets you specify a loading indicator for a part of the component tree
- **Error Boundary** - A React component that catches JavaScript errors anywhere in their child component tree
- **Preact** - A fast 3kB alternative to React with the same modern API
- **Vue.js** - A JavaScript framework for building user interfaces
- **Angular** - A web framework that empowers developers to build fast, reliable applications
- **Svelte** - A UI framework that uses a compiler to let you write breathtakingly concise components that do minimal work in the browser, using languages you already know — HTML, CSS and JavaScript
- **Ember.js** - A framework for ambitious web developers
- **HTML-First Frameworks**
 - **htmx** - A library that allows you to access AJAX, CSS Transitions, WebSockets and Server Sent Events directly in HTML, using attributes
 - **htm** - A JSX alternative using standard tagged templates, with no transpiler necessary
- **Framework-agnostic Core Libraries**
 - **TanStack** - A collection of high-quality, framework-agnostic open-source libraries for web development
 - **TanStack Query** - A powerful asynchronous state management for TS/JS,

React, Solid, Vue, Svelte and Angular

- [TanStack Router](#) - A powerful, type-safe, and framework-agnostic router for building modern web applications
- [TanStack Table](#) - A headless UI for building powerful tables & datagrids for TS/JS, React, Vue, Solid and Svelte
- [TanStack Form](#) - A type-safe and framework-agnostic form state management for React, Vue, Solid, and Svelte
- [TanStack Virtual](#) - A headless UI for virtualizing large lists and grids in React, Vue, Svelte, Solid and JS

State, Routing & Logic

- State Management
 - [Redux](#) - A JS library for predictable and maintainable global state management
 - [React-Redux](#) - The official React binding for Redux
 - [Zustand](#) - A small, fast, and scalable barebones state-management solution using simplified flux principles
 - [Recoil](#) - A state management library for React
 - [XState](#) - A library for creating, interpreting, and executing finite state machines and statecharts
- Routing
 - [React Router](#) - A user-obsessed, standards-focused, multi-strategy router you can deploy anywhere
- Syntax & Templating
 - [JSX](#) - A syntax extension for JavaScript that lets you write HTML-like markup inside a JavaScript file
 - [MDX](#) - An authorable format that lets you seamlessly write JSX in your Markdown documents
- WASM Runtimes
 - [PyScript](#) - A free Open Source Software (OSS) that facilitates the creation, deployment, and sharing of Python applications

Styling & UI Components

- CSS Ecosystem
 - Frameworks and UI Kits

- [Bootstrap](#) - The world's most popular front-end open source toolkit
- [Tailwind CSS](#) - A utility-first CSS framework packed with classes
- [Oat](#) - An ultra-lightweight, semantic, zero-dependency HTML UI component library that provides minimal, standards-based CSS and JS
- Tailwind Component Libraries
 - [daisyUI](#) - The most popular component library for Tailwind CSS
- CSS-in-JS
 - [Emotion](#) - A library designed for writing css styles with JavaScript
 - [Linaria](#) - A zero-runtime CSS in JS library
- Preprocessors
 - [Sass language](#) - A stylesheet language that's compiled to CSS
- Transforms
 - [CSS Transforms 1](#) - A CSS module that allows elements to be transformed in two-dimensional space
 - [CSS Transforms 2](#) - A CSS module that allows elements to be transformed in three-dimensional space
- UI Component Libraries
 - [templUI](#) - A growing collection of beautifully designed UI components for Go and templ
 - [Material UI](#) - An open-source React component library that implements Google's Material Design
 - [Chakra UI](#) - A component system for building products with speed
 - [Vuetify](#) - A no design skills required Open Source UI Library with beautifully handcrafted Vue Components
- Specialized UI Widgets
 - Rich Text Editors
 - [Tiptap](#) - The headless and open source editor framework designed for web developers
 - Interaction & Media
 - [Swiper.js](#) - The most modern mobile touch slider with hardware accelerated transitions and amazing native behavior
 - [Hammer.js](#) - A javascript library for multi-touch gestures
 - Canvas & Whiteboarding

- [tldraw](#) - A React-based SDK for building whiteboards, diagrams, and canvas tools with a high-performance web canvas

Build & Development Tooling

- Development Environments
 - [Storybook](#) - A frontend workshop for building UI components and pages in isolation
- Bundlers
 - [Vite](#) - A build tool that aims to provide a faster and leaner development experience for modern web projects
 - [Parcel](#) - The zero configuration build tool
 - [webpack](#) - A static module bundler for modern JavaScript applications
 - [Rspack](#) - A high performance JavaScript bundler written in Rust
 - [Rsbuild](#) - The Rspack-based web build tool
- Transpilers
 - [babel](#) - A JavaScript compiler
- Minifiers
 - [JSMIn](#) - A minification tool that removes comments and unnecessary whitespace from JavaScript files
- Linters & Formatters
 - [Biome](#) - A fast formatter and linter for JavaScript, TypeScript, JSX, TSX, JSON, HTML, CSS and GraphQL that provides a cohesive toolchain for web projects
 - [Knip](#) - A tool that finds and fixes unused dependencies, exports and files in JavaScript and TypeScript projects

Full-stack & Static Site Frameworks

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Front-end System Development
- 3. Technology > 3.1 Software Development > Back-end System Development

Full-stack Frameworks

- JS/TS Full-stack Frameworks

- [Next.js](#) - A React framework for building full-stack web applications
- [Nuxt.js](#) - A free and open-source framework with an intuitive and extendable way to create type-safe, performant and production-grade full-stack web applications and websites with Vue.js
- [Astro](#) - The web framework for content-driven websites
- [Fresh](#) - A next generation web framework, built for speed, reliability, and simplicity
- Rust Full-Stack Frameworks
 - [Leptos](#) - A cutting-edge Rust framework for the modern web

Static Site Generators

- [Docusaurus](#) - A static-site generator. It builds a single-page application with fast client-side navigation, leveraging the full power of React to make your site interactive
- [mdBook](#) - A utility to create modern online books from Markdown files
- [VuePress](#) - A Vue-powered Static Site Generator
- [Hugo](#) - The world's fastest framework for building websites
 - [Docsy](#) - A Hugo theme for technical documentation sites, providing easy site navigation, structure, and more
- [Jekyll](#) - A simple, blog-aware, static site generator perfect for personal, project, or organization sites
- [Eleventy](#) - A simpler static site generator written in JavaScript
- [Sphinx](#) - A tool that makes it easy to create intelligent and beautiful documentation
- [MkDocs](#) - A fast, simple and downright gorgeous static site generator that's geared towards building project documentation
 - [Material for MkDocs](#) - A powerful and beautiful theme for the MkDocs static site generator
- [Nanoc](#) - A static-site generator, fit for building anything from a small personal blog to a large corporate website
- [gitmal](#) - A static page generator designed for Git repositories

Headless CMS

- Cloud-native & API-first CMS
 - [Contentful](#) - A headless content management system that provides a content-

first approach to building digital products

- [Strapi](#) - The leading open-source headless CMS
- [Sanity](#) - A platform for structured content that lets you build better digital experiences

Backend Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Back-end System Development

API Architectural Styles

- [REST](#) - A software architectural style that was created to guide the design and development of the architecture for the World Wide Web
- [SOAP \(legacy\)](#) - A messaging protocol specification for exchanging structured information in the implementation of web services
- [GraphQL](#) - A query language for APIs and a runtime for fulfilling those queries with your existing data
- [gRPC](#) - A modern open source high performance Remote Procedure Call (RPC) framework that can run in any environment
- [json-rpc](#) - A stateless, light-weight remote procedure call (RPC) protocol
- [Webhook](#) - A method of augmenting or altering the behavior of a web page or web application with custom callbacks

Backend Frameworks

- JS/TS Backend Frameworks
 - [Fastify](#) - A fast and low-overhead web framework for Node.js, designed for optimal performance and developer experience
 - [Express.js](#) - A minimal and flexible Node.js web application framework
 - [Koa](#) - A new web framework designed by the team behind Express
 - [Nest.js](#) - A progressive Node.js framework for building efficient, reliable and scalable server-side applications
 - [Hono](#) - A small, simple, and ultrafast web framework for the Edges
- API Tools
 - [tRPC](#) - A tool that allows you to easily build & consume fully typesafe APIs

without schemas or code generation

- Go Backend Frameworks
 - [Echo](#) - A high performance, extensible, minimalist Go web framework
 - [Fiber](#) - An Express inspired web framework built on top of Fasthttp, the fastest HTTP engine for Go, designed to ease development with performance in mind
 - [Gin Web Framework](#) - A web framework written in Go
 - [Gorilla web toolkit](#) - A helpful toolkit that provides useful, composable packages for writing HTTP-based applications
 - [Yokai](#) - A simple, modular and observable Go framework for backend applications
 - [Kratos](#) - A lightweight Go framework for building cloud-native microservices, with APIs for transport, middleware, registry, configuration, logging, and code generation
- Python Backend Frameworks & Servers
 - [WSGI](#) - The Web Server Gateway Interface
 - [Gunicorn](#) - A Python WSGI HTTP Server for UNIX
 - [Flask](#) - A lightweight WSGI web application framework
 - [ASGI](#) - A spiritual successor to WSGI, the long-standing Python standard for compatibility between web servers, frameworks, and applications
 - [Uvicorn](#) - A lightning-fast ASGI server implementation for Python, using uvloop and httptools for high performance
 - [Hypercorn](#) - An ASGI and WSGI web server based on the sans-io hyper, h11, h2, and wsproto libraries with support for HTTP/1, HTTP/2, and HTTP/3
 - [FastAPI](#) - A modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints
 - [SlowAPI](#) - A small library to rate limit your ASGI applications
- Ruby Backend Frameworks & Servers
 - [Ruby on Rails](#) - A web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern
 - [Rack](#) - A modular Ruby web server interface
 - [Puma](#) - A fast, concurrent web server for Ruby & Rack
 - [Falcon](#) - A multi-process, multi-fiber rack-compatible HTTP server built on top of async, async-container and async-http

- [Sinatra](#) - A DSL for quickly creating web applications in Ruby with minimal effort
- [Sidekiq](#) - A simple, efficient background processing tool for Ruby
- [Shrine](#) - A file attachment toolkit for Ruby applications
- Perl Backend Frameworks (legacy)
 - Classic CGI
 - [mod_cgi](#) - A module for the execution of CGI scripts
 - [CGI.pm](#) - A module to handle Common Gateway Interface requests and responses
 - Fast CGI
 - [mod_fcgid](#) - A high performance alternative to mod_cgi or mod_cgid
 - [FCGI.pm](#) - A module for FastCGI applications
- Java Backend Frameworks
 - [Jakarta EE](#) - A set of specifications that define Java APIs for enterprise software development
 - [Apache Tomcat](#) - An open-source web server and servlet container
 - [Spring](#) - A project that makes Java simple, modern, productive, reactive, and cloud-ready
 - [Spring Boot](#) - A tool that takes an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss
- .NET Backend Frameworks
 - [ASP.NET](#) - A free, cross-platform, open source framework for building web apps and services with .NET and C#
- Elixir Backend Frameworks
 - [Phoenix](#) - A web framework for building rich, interactive web applications quickly with less code and fewer moving parts, used to craft APIs, HTML5 apps, and more at scale
- GraphQL Servers
 - [Apollo Server](#) - An open-source, spec-compliant GraphQL server that's compatible with any GraphQL client

Web Infrastructure

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

Web Server & Proxy

- Web Servers & Reverse Proxy Servers
 - [NGINX](#) - An open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more
 - [Apache HTTP Server](#) - A project to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows
 - [Caddy](#) - A powerful, extensible platform to serve your sites, services, and apps, written in Go
 - [HAProxy](#) - A free, very fast and reliable reverse-proxy offering high availability, load balancing, and proxying for TCP and HTTP-based applications
 - [nodejs http-server](#) - A simple static HTTP server
 - [goshs](#) - A feature-rich single-binary file server for red teamers and developers supporting HTTP/S, WebDAV, SFTP, SMB, LDAP/S, NTLM hash capture, DNS/SMTP callbacks, TLS, authentication, and share links
- API Management
 - [Unkey](#) - An open-source API management platform designed to help developers secure, manage, and scale their APIs
 - [Kong API gateway](#) - A lightweight, fast, and flexible cloud-native API gateway
 - [Azure API Management](#) - A hybrid, multicloud management platform for APIs across all environments
 - [Amazon API Gateway](#) - A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale
 - [Google Cloud Apigee](#) - The platform for developing and managing API services
 - [Gravitee](#) - A unified API visibility and governance platform that provides a single pane of glass for managing, securing, and governing APIs across any infrastructure

CDN & Edge Computing

- Concepts
 - [Web cache](#) - An information technology for the temporary storage (caching) of web documents, such as HTML pages and images, to reduce bandwidth usage, server load, and perceived lag
 - [Content delivery network](#) - A geographically distributed network of proxy

servers and their data centers

- [Point of presence](#) - An artificial demarcation point or interface point between communicating entities
- Forward Proxy Servers
 - [Squid](#) - A caching proxy for the Web supporting HTTP, HTTPS, FTP, and more
- CDN Providers
 - [Cloudflare](#) - A global network designed to make everything you connect to the Internet secure, private, fast, and reliable
 - [Cloudflare Workers](#) - A serverless execution environment that allows you to create entirely new applications or augment existing ones without configuring or maintaining infrastructure
 - [Cloudflare Workers Bindings](#) - A mechanism that allows your Worker to interact with resources on the Cloudflare Developer Platform, providing better performance and fewer restrictions than REST APIs for accessing resources from Workers
 - [Amazon CloudFront](#) - A content delivery network (CDN) service built for high performance, security, and developer convenience
 - [Lambda@Edge](#) - A feature of Amazon CloudFront that lets you run code closer to users of your application
 - [Google Cloud CDN](#) - A content delivery network (CDN) that accelerates delivery of your web and video content
 - [Azure Front Door](#) - A modern cloud content delivery network (CDN) that provides a secure and scalable entry point for fast delivery of your global web applications and content
- JAMstack Hosting
 - [GitLab Pages](#) - A feature that allows you to publish static websites directly from a repository in GitLab
 - [Cloudflare Pages](#) - A JAMstack platform for frontend developers to collaborate and deploy websites

Decentralized Web

Relevant DSS-P Skills

- 3. Technology > 3.2 Digital Technology > Other Advanced Technologies

Blockchain Technology

- [Web3](#) - An idea for a new iteration of the World Wide Web which incorporates concepts such as decentralization, blockchain technologies, and token-based economics
- [Blockchain](#) - A distributed ledger with growing lists of records
 - [Hashcash](#) - A proof-of-work system used to limit email spam and denial-of-service attacks
 - [Proof of work](#) - A form of cryptographic proof in which one party proves to others that a certain amount of a specific computational effort has been expended
- [Smart contract](#) - A computer program or transaction protocol designed to automatically execute, control, or document events and actions according to contract terms
- [Bitcoin](#) - A decentralized digital currency that can be transferred on the peer-to-peer bitcoin network
- [Ethereum](#) - A global, decentralized network that provides direct ownership of digital assets, data, and identity without requiring permission from any central authority
- [Non-fungible token](#) - A unique digital identifier that is recorded on a blockchain and is used to certify ownership and authenticity
- [Decentralized autonomous organization](#) - A member-owned community without centralized leadership managed by decentralized computer programs with voting and finances handled through a blockchain
- [Solidity](#) - A programming language for implementing smart contracts on various blockchain platforms, most notably Ethereum
- [Web3.js](#) - A TypeScript/JavaScript library that enables developers to connect to and interact with Ethereum and other EVM-compatible blockchains
- [ethers.js](#) - A simple, compact and complete JavaScript library for all your Ethereum needs
- [MetaMask](#) - A crypto wallet that enables users to buy, sell, swap, and store cryptocurrencies while maintaining control over their data and assets
- [WalletConnect](#) - An open-source protocol that establishes encrypted connections between mobile cryptocurrency wallets and desktop-based decentralized applications
- [Hardhat](#) - A development environment for Ethereum and EVM-compatible blockchains that helps developers compile, deploy, test, and debug Solidity smart contracts

Decentralized Social

- [ActivityPub](#) - A decentralized social networking protocol based on the ActivityStreams 2.0 data format
- [AT Protocol](#) - An open data network for building social applications where users own their identities and content is represented as interlinked JSON records
- [Fediverse](#) - An ensemble of interconnected servers that are used for web publishing and file hosting, but which can communicate with each other
- [Mastodon](#) - A free, open-source, decentralized social media platform that puts users in control of their feeds without algorithms or ads, allowing independent servers to interoperate through the ActivityPub protocol
- [Bluesky](#) - A microblogging social media service and a public benefit corporation based in the United States
- [Nostr](#) - An open, decentralized social protocol that uses cryptographic signatures to enable censorship-resistant communication across multiple independent servers called relays
- [Matrix](#) - An open standard and communication protocol for real-time communication that enables seamless communication between different service providers
- [PeerTube](#) - A free, open-source tool for creating independent video hosting platforms that connect to form a decentralized network, offering an alternative to centralized services
- [Lemmy](#) - A decentralized discussion platform that allows users to control their experience without corporate tracking or advertising
- [Diaspora](#) - A nonprofit, user-owned, distributed social network consisting of independently owned nodes called pods that interoperate to form the network
- [Secure Scuttlebutt](#) - A decentralized social network platform that enables local community development free from corporate data harvesting

Development & Testing Tools

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- 3. Technology > 3.1 Software Development > Team Development

Web/HTTP Clients

- HTTP CLI Tools
 - [cURL](#) - A command line tool and library for transferring data with URLs
 - [Wget](#) - A free software package for retrieving files using HTTP, HTTPS, FTP and FTPS
 - [curlie](#) - The power of curl, the ease of use of httpie
 - [hurl](#) - A command line tool that runs HTTP requests defined in a simple plain text format
 - [httpie cli](#) - A simple yet powerful command-line HTTP and API testing client for the API era
 - [wuzz](#) - An interactive CLI tool for HTTP inspection
 - [httptap](#) - A tool to view the HTTP and HTTPS requests made by any Linux program
- HTTP Client Libraries
 - [Python Requests](#) - An elegant and simple HTTP library for Python, built for human beings
 - [JS Axios](#) - A promise-based HTTP Client for node.js and the browser
 - [Go Resty](#) - A simple HTTP and REST client library for Go
 - [Go FastHTTP](#) - A fast HTTP package for Go
 - [Surf](#) - An advanced Go HTTP client with Chrome/Firefox browser impersonation, HTTP/3 with QUIC fingerprinting, JA3/JA4 TLS emulation, and anti-bot bypass
 - [Typhoeus](#) - A library that wraps libcurl in order to make fast and reliable requests
 - [Ruby Net](#) - A collection of classes that implement client-side internet protocols
 - [httpx](#) - An HTTP client library for the Ruby programming language
 - [wreq-ruby](#) - An easy and powerful Ruby HTTP client with advanced browser fingerprinting that accurately emulates various browsers with precise TLS/HTTP2 signatures
 - [Rust request](#) - An ergonomic, async HTTP client
- GraphQL Libraries
 - [URQL](#) - The highly customizable and versatile GraphQL client for React, Svelte, Vue, or plain JavaScript
- API Testing Platforms

- [Bruno](#) - A Git-integrated, fully offline, and open-source API client
- [Postman/Newman](#) - An API platform for building and using APIs
- Classic Web Automation
 - [Mechanize](#) - A module that helps you automate interaction with a website
 - [Mechanize \(Ruby\)](#) - A ruby library that makes automated web interaction easy

Web Debugging Tools

- [Chrome DevTools](#) - A set of web developer tools built directly into the Google Chrome browser
- [Firefox Developer Tools](#) - A set of web developer tools built into Firefox that allow you to examine, edit, and debug HTML, CSS, and JavaScript
- [React Developer Tools](#) - A browser extension and standalone debugger that allows developers to inspect React components, edit props and state, and identify performance problems in React applications
- [Vue.js devtools](#) - A browser extension for debugging Vue.js applications that provides component inspection and state management debugging
- [Redux DevTools](#) - A development tool that provides power-ups for Redux development workflow, including hot reloading, action replay, and customizable UI
- [Lighthouse](#) - An open-source, automated tool that helps improve web page quality by auditing performance, accessibility, SEO, and best practices
- [Fiddler](#) - A free web debugging proxy for any browser, system or platform
- [Charles Proxy](#) - An HTTP proxy/monitor that enables developers to view all HTTP and SSL/HTTPS traffic between their machine and the Internet, including requests, responses, and headers
- [mitmproxy](#) - A free and open source interactive HTTPS proxy that can intercept, inspect, modify, and replay web traffic for debugging, testing, and penetration testing purposes
- [Requestly](#) - An HTTP interceptor that allows developers to modify URLs, headers, and API responses in real-time for debugging and testing

Web Test Automation Frameworks

- Browser Automation & Testing
 - [Puppeteer](#) - A Node.js library which provides a high-level API to control Chrome/Chromium over the DevTools Protocol
 - [Playwright](#) - A framework for reliable end-to-end testing for modern web apps

with a single API for Chromium, Firefox, and WebKit

- [Playwright for Go](#) - A Go library to automate Chromium, Firefox and WebKit with a single API
- [Cypress](#) - An open-source, JavaScript-based testing framework that enables developers to write, run, and debug end-to-end and component tests directly in the browser for modern web applications
- [WebDriver](#) - A remote control interface that enables introspection and control of user agents
 - [Selenium WebDriver](#) - A tool that drives a browser natively, as a user would, either locally or on a remote machine
 - [WebDriver BiDi](#) - The BiDirectional WebDriver Protocol, a mechanism for remote control of user agents
- [Selenium IDE](#) - An open source record and playback test automation for the web
- [Chrome DevTools Protocol \(CDP\)](#) - A low-level API that allows external tools to instrument, inspect, debug, and profile Chromium-based browsers
- [Karma](#) - A test runner that spawns a web server and executes source code against test code for each of the connected browsers
- Supporting Tools
 - [Chrome for Testing](#) - A new flavor of Chrome that specifically targets web app testing and automation use cases
- Accessibility Testing
 - [axe-core](#) - An accessibility testing engine for websites and other HTML-based user interfaces
- AI-powered Web Automation
 - [browser-use](#) - An open-source Python library that allows AI agents to interact with web browsers using natural language
- Web Scraping
 - [Crawlee](#) - A web scraping and browser automation library
 - [BeautifulSoup](#) - A Python library designed for quick turnaround projects like screen-scraping
 - [Scrapy](#) - An open source and collaborative framework for extracting the data you need from websites
 - [Colly](#) - A Golang framework for building web scrapers
 - [Katana](#) - A next-generation crawling and spidering framework

- [Trafalatura](#) - A Python package and command-line tool to gather text on the Web

03 - Cloud & Cloud-Native Computing

Cloud Computing

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

Computing & Storage (IaaS)

- [Amazon EC2](#) - A web service that provides secure, resizable compute capacity in the cloud
- [Amazon EBS](#) - An easy-to-use, high-performance block storage service designed for use with Amazon Elastic Compute Cloud
- [Azure Virtual Machines](#) - A service to provision Windows and Linux virtual machines in seconds
- [Azure Disk Storage](#) - A high-performance, durable block storage for Azure Virtual Machines
- [Google Cloud Compute Engine](#) - A customizable compute service that lets you create and run virtual machines on Google's infrastructure

Networking

- [Amazon VPC](#) - A service that lets you launch AWS resources in a logically isolated virtual network that you define
- [Amazon ELB](#) - A service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions
- [Azure Virtual Network](#) - The fundamental building block for your private network in Azure with access to high-performance networking
- [Azure Load Balancer](#) - A service that allows you to distribute traffic to your backend virtual machines
- [Azure Application Gateway](#) - A platform-managed, scalable, and highly available application delivery controller as a service
- [Google Cloud VPC](#) - A virtual version of a physical network that is implemented inside of Google's production network by using Andromeda

- [Cloud Load Balancing](#) - A fully distributed, software-defined, managed service for all your traffic

Application Hosting Platform (PaaS)

- [Azure App Service](#) - An HTTP-based service for hosting web applications, REST APIs, and mobile back ends
- [AWS Elastic Beanstalk](#) - An easy-to-use service for deploying and scaling web applications and services
- [Google Cloud App Engine](#) - A fully managed, serverless platform for developing and hosting web applications at scale
- [Vercel](#) - A frontend cloud platform that provides the developer experience and infrastructure to build, deploy, and scale the web
- [Netlify](#) - A composable web platform that enables enterprises and teams to build, deploy, and scale modern web experiences on a global edge network
- [Coolify](#) - An open-source & self-hostable alternative to Vercel, Heroku, Netlify and Railway

Cloud Command Line Interfaces

- [AWS CLI](#) - A unified tool to manage your AWS services
- [Azure CLI](#) - A cross-platform command-line tool for managing Azure resources with interactive commands or scripts
- [Azure Developer CLI \(azd\)](#) - An open-source tool that accelerates your path from a local development environment to Azure
- [Google Cloud CLI \(gcloud\)](#) - A set of tools to create and manage Google Cloud resources and services
- [Vercel CLI](#) - A command-line interface used to manage and configure Vercel Projects, deploy apps, and replicate the deployment environment locally
- [Netlify CLI](#) - A command-line interface to configure continuous deployment, run a local development server, and deploy sites on Netlify

Cloud Emulators

- [LocalStack](#) - A fully functional local cloud stack to develop and test your cloud and serverless apps offline

Private Cloud & On-Premises IaaS

- [OpenStack](#) - An open source cloud computing platform that provides infrastructure as a service (IaaS) for building and managing public and private clouds
- [OpenNebula](#) - The Enterprise Cloud and Virtualization Platform that delivers end-to-end coverage, vendor neutrality, and comprehensive support across cloud and virtualization management, hypervisor operations, and Kubernetes orchestration

Cloud Architecture Frameworks

- [AWS Well-Architected Framework](#) - A framework describing key concepts, design principles, and architectural best practices for designing and running workloads in the cloud
- [Azure Architecture Center](#) - A set of guidance, patterns, and best practices for building secure, high-performing, resilient, and efficient infrastructure on Azure
- [Azure Well-Architected Framework](#) - A set of quality-driven tenets, architectural decision points, and review tools intended to help solution architects build a technical foundation for their workloads

Configuration as Code

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

Infrastructure as Code (IaC)

- [Hashicorp Terraform](#) - An infrastructure as code tool that lets you build, change, and version infrastructure safely and efficiently
- [OpenTofu](#) - An open-source, community-driven fork of Terraform that provides a stable, drop-in replacement for building and managing infrastructure
- [Pulumi](#) - An infrastructure as code platform that allows you to use familiar programming languages and tools to build, deploy, and manage cloud infrastructure

AI-driven Infrastructure

- [Spacelift Intent](#) - An AI-powered tool that allows users to provision and manage cloud infrastructure using natural language

Configuration Management & Automation

- [Ansible](#) - An open source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes
- [SaltStack](#) - A Python-based, open-source software for event-driven IT automation, remote task execution, and configuration management
- [Rudder](#) - An open-source, continuous configuration and compliance platform for IT infrastructure automation
- [cloud-init](#) - The standard for customising cloud instances

Image Building

- [Hashicorp Packer](#) - A tool for creating identical machine images for multiple platforms from a single source configuration

Ecosystem & Vendor Tools

- Terraform/OpenTofu Ecosystem
 - [Terraform/OpenTofu Provider: Core Functions](#) - A Terraform/OpenTofu provider for performing core functions
 - [Terragrunt](#) - A thin wrapper that provides extra tools for keeping your configurations DRY, working with multiple Terraform modules, and managing remote state
 - [TerraTest](#) - A Go library that provides patterns and helper functions for testing infrastructure
 - [Atmos](#) - A universal tool for DevOps and Cloud Engineering that orchestrates workflows and simplifies the management of infrastructure
 - [GitLab-managed Terraform/OpenTofu state](#) - A feature that allows you to store your Terraform state files in GitLab
 - [tf.libsonnet](#) - A collection of Jsonnet libraries for generating Terraform code
 - [terraform-docs](#) - A utility to generate documentation from Terraform modules in various output formats
 - [Terraformer](#) - A CLI tool to generate terraform files from existing infrastructure
 - [Atlantis](#) - A self-hosted go-lang application that listens for Terraform pull request events via webhooks
- Vendor-specific Tools

- [AWS CloudFormation](#) - A service that helps you model and set up your Amazon Web Services resources
- [AWS CDK](#) - An open source software development framework to define your cloud application resources using familiar programming languages
- [AWS SAM](#) - An open-source framework for building serverless applications
- [SST](#) - A framework that makes it easy to build full-stack applications on AWS with a specialized focus on serverless and event-driven architectures
- [Azure Resource Manager](#) - The deployment and management service for Azure
 - [Bicep language](#) - A domain-specific language (DSL) that uses declarative syntax to deploy Azure resources
 - [Azure Resource Graph](#) - A powerful management tool to query, explore, and analyze your cloud resources at scale

Containerization

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [Containerization](#) - A form of operating-system-level virtualization
- Linux Distros for Containers
 - [Alpine Linux](#) - A security-oriented, lightweight Linux distribution based on musl libc and busybox
 - [apk-tools](#) - A package manager originally built for Alpine Linux
 - [Fedora CoreOS](#) - An automatically updating, minimal operating system for running containerized workloads securely and at scale
 - [Flatcar Container Linux](#) - An immutable Linux distribution for containers
- Utilities in Containers
 - [busybox](#) - A single small executable that combines tiny versions of many common UNIX utilities
- Standards
 - [The Open Container Initiative \(OCI\)](#) - An open governance structure for the express purpose of creating open industry standards around container formats and runtimes
 - [Compose Specification](#) - A developer-focused standard for defining cloud and platform agnostic container-based applications

- [Development Containers](#) - An open specification for enriching containers with development-specific settings, tools, and configuration

Engines & Runtimes

- Container Engines
 - [Docker Engine](#) - An open source containerization technology for building and containerizing your applications
 - [Docker Rootless mode](#) - A feature that allows the Docker daemon and containers to run as a non-root user, mitigating potential vulnerabilities
 - [podman](#) - A powerful container engine for building, managing, and running containers and pods
 - [podman-static](#) - Alpine-based container images and statically linked (rootless) binaries for Linux
 - [WSL Container](#) - A built-in container engine for Windows Subsystem for Linux that provides a CLI ([wslc.exe](#)) and API for building, running, and managing Linux containers on Windows
- Container Runtimes
 - [containerd](#) - An industry-standard container runtime with an emphasis on simplicity, robustness and portability
 - [nerdctl](#) - A Docker-compatible CLI for containerd
 - [ctr](#) - An unsupported debug and administrative client for interacting with the containerd daemon
 - [CRI-O](#) - An implementation of the Kubernetes CRI (Container Runtime Interface) to enable using OCI (Open Container Initiative) compatible runtimes
 - [cri-tools](#) - A set of tools for CRI
- OCI Runtimes
 - [runc](#) - A CLI tool for spawning and running containers according to the OCI specification
 - [crun](#) - A fast and lightweight fully featured OCI runtime and C library for running containers

Image Management

- Image Building Tools
 - [Docker Build](#) - A part of the Docker Engine that automates the process of creating a Docker image from a Dockerfile and a context

- [buildah](#) - A tool that facilitates building Open Container Initiative (OCI) container images
- [podman build](#) - A command that constructs OCI-compatible container images by interpreting instructions from a Containerfile or Dockerfile, leveraging Buildah for the underlying operations
- [Kaniko](#) - A tool to build container images from a Dockerfile, inside a container or Kubernetes cluster
- [Cloud Native Buildpacks](#) - A tool that transforms your application source code into images that can run on any cloud, without a Dockerfile
- Image Inspection & Management Tools
 - [skopeo](#) - A command line utility that performs various operations on container images and image repositories
 - [dive](#) - A tool for exploring a docker image, layer contents, and discovering ways to shrink the size of your Docker/OCI image
 - [regclient](#) - A suite of command-line tools (regctl, regsync, regbot) for managing and inspecting OCI registries and images, supporting advanced features like multi-platform images and mirroring
- Container Registries
 - [GitLab Container Registry](#) - A secure and private registry for Docker images
 - [Project Quay](#) - An open-source, container-native image registry designed for building, organizing, distributing, and deploying containers
 - [Docker Hub](#) - A cloud-based registry service that allows developers and teams to store, share, and distribute Docker container images
 - [Amazon ECR](#) - A fully managed container registry that makes it easy to store, manage, share, and deploy your container images and artifacts
 - [Azure Container Registry](#) - A private registry for managing container images and related artifacts
 - [Harbor](#) - An open source registry that secures artifacts with policies and role-based access control

Environment & Management

- Container Management Tools
 - [Podman Desktop](#) - The best free and open source tool for developers to work with containers and Kubernetes, simplifying container management, streamlining Kubernetes workflows, and transitioning from local development to production with ease

- [lazydocker](#) - A terminal UI for both docker and docker-compose
- [Docker Compose](#) - A tool for defining and running multi-container Docker applications
- Development Environment Provisioning
 - [devcontainers CLI](#) - A reference implementation for the specification that can create and configure a dev container from a devcontainer.json, providing commands for building, running, and managing development containers across different infrastructures
 - [DevPod](#) - A client-only tool that enables dev-environments-as-code using the open standard devcontainer.json format, supporting any infrastructure including local machines, Kubernetes clusters, and cloud providers
- Local Environment Provisioners (for Mac)
 - [Colima](#) - A tool that provides container runtimes on macOS (and Linux) with minimal setup
 - [Lima](#) - A tool that launches Linux virtual machines with automatic file sharing and port forwarding

WebAssembly

Relevant DSS-P Skills

- 3. Technology > 3.2 Digital Technology > Other Advanced Technologies
- Standards
 - [WebAssembly](#) - A binary instruction format for a stack-based virtual machine
 - [WebAssembly System Interface \(WASI\)](#) - A modular system interface for WebAssembly
 - [WASIX](#) - The long term stabilization and support of the existing WASI ABI plus additional non-invasive syscall extensions
 - [WebAssembly Component Model](#) - A binary interface standard letting modules interoperate through typed, language-agnostic interfaces, developed as part of WASI Preview 2
- Runtimes
 - [wazero](#) - The only zero dependency WebAssembly runtime written in Go
 - [Wasmtime](#) - A fast and secure runtime for WebAssembly
 - [Wasmer](#) - A blazing fast and secure WebAssembly runtime that enables incredibly lightweight containers to run anywhere

- [WasmEdge](#) - A lightweight, high-performance, and extensible runtime for cloud native, edge, and decentralized applications
- [WebAssembly Micro Runtime \(WAMR\)](#) - A lightweight standalone runtime with small footprint and highly configurable features for embedded, IoT, edge, and Trusted Execution Environment use
- Toolchains & Languages
 - [Emscripten](#) - A complete compiler toolchain to WebAssembly, using LLVM, with a special focus on speed, size, and the Web platform
 - [AssemblyScript](#) - A TypeScript-like language for WebAssembly
 - [TinyGo](#) - A Go compiler for small places, bringing the language to embedded systems and WebAssembly
 - [Binaryen](#) - A compiler and toolchain infrastructure library for WebAssembly, written in C++
- Cloud & Edge Platforms
 - [wasmCloud](#) - An open source CNCF project to build polyglot applications composed of reusable WebAssembly components across cloud, Kubernetes, and edge
 - [Fermion Spin](#) - The developer tool for building WebAssembly microservices and web applications

Kubernetes

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [Kubernetes](#) - An open-source system for automating deployment, scaling, and management of containerized applications
- Master node
 - kube-apiserver - Responsible for API services
 - kube-scheduler - Responsible for scheduling
 - kube-controller-manager - Responsible for container orchestration
- Compute node
 - kubelet - watches the API server for pods on that node and makes sure they are running
 - cAdvisor - collects metrics about pods running on that particular node

- kube-proxy - watches the API server for pods/services changes in order to maintain the network up to date
- container runtime - responsible for managing container images and running containers on that node
- Interface Standards
 - CNI (Container Networking Interface)
 - CSI (Container Storage Interface)
 - CRI (Container Runtime Interface)

Core Concepts & Components

- K8s Internals
 - [Workloads](#) - The objects you use to manage and run your containers on the cluster
 - Pod
 - [assignment](#) - The process of constraining a Pod so that it is restricted to run on particular nodes, or to prefer to run on particular nodes
 - [taint and toleration](#) - A mechanism that allows you to ensure that pods are not placed on inappropriate nodes
 - [lifecycle](#) - The lifecycle of a Pod
 - [liveness probe](#) - A probe the kubelet uses to know when to restart a container
 - requests and limits
 - eviction
 - Deployment, ReplicaSet, StatefulSet, DaemonSet
 - [Kubernetes network model](#) - A set of fundamental requirements and principles for networking in a Kubernetes cluster
 - Service, Ingress, Ingress Controllers
 - [Storage](#) - A powerful volume subsystem with an API that abstracts how storage is provided and consumed
 - PersistentVolume, PVC, StorageClass
 - [Configuration](#) - A range of mechanisms that let you inject configuration data into the Pods that run your applications
 - Secret, ConfigMap

- Security & Policy
 - [Kubernetes RBAC](#) - A method of regulating access to computer or network resources based on the roles of individual users within an enterprise
 - [PodDisruptionBudget](#) - An object that limits the number of concurrent disruptions that your application experiences, allowing for high availability
 - [Security context](#) - A definition of privilege and access control settings for a Pod or Container
- Autoscaling
 - [HPA](#) - The component that automatically scales the number of Pods in a replication controller, deployment, replica set or stateful set based on observed CPU utilization
 - [Cluster Autoscaler](#) - A tool that automatically adjusts the size of the Kubernetes cluster
 - [Karpenter](#) - A flexible, high-performance Kubernetes cluster autoscaler

Operations & Management

- K8s Operators
 - [Prometheus Operator](#) - The operator that creates/configures/manages Prometheus clusters atop Kubernetes
 - [kube-prometheus](#) - A collection of Kubernetes manifests, Grafana dashboards, and Prometheus rules combined with documentation and scripts to provide easy to operate end-to-end Kubernetes cluster monitoring
 - [Grafana Operator](#) - A Kubernetes operator that streamlines the management of Grafana instances, dashboards, and data sources through custom resources in Kubernetes and OpenShift environments
 - [OpenTelemetry Operator](#) - An implementation of a Kubernetes Operator for OpenTelemetry
 - [Elastic Cloud on Kubernetes \(ECK\)](#) - The official operator for the Elastic Stack on Kubernetes
 - [Rook](#) - An open source cloud-native storage orchestrator for Kubernetes
- Dashboards
 - [k9s](#) - A terminal based UI to interact with your Kubernetes cluster
 - [KDash](#) - A simple terminal dashboard for Kubernetes built with Rust
 - [Seabird](#) - The native desktop app that simplifies working with Kubernetes

- [Headlamp](#) - A user-friendly Kubernetes UI focused on extensibility

CLI & Local Environments

- CLI Plugin Management
 - [Krew](#) - The plugin manager for kubectl command-line tool
 - [kubectl-node-shell](#) - A kubectl plugin to run a root shell on a node
 - [kubectl-tree](#) - A kubectl plugin to explore ownership relationships between Kubernetes objects
 - [kubectl-pod-inspect](#) - A kubectl plugin to view pod and container status at a glance
 - [kubepug](#) - A pre-flight checking tool for Kubernetes APIs
 - [rakkess](#) - A kubectl plugin to show an access matrix for all available resources
 - [ketall](#) - A kubectl plugin to get all resources
- Local K8s Tools
 - [Minikube](#) - A tool that lets you run Kubernetes locally
 - [Kind](#) - A tool for running local Kubernetes clusters using Docker container "nodes"

Ecosystem & Extensions

- Application Packaging & Configuration
 - [Helm](#) - The package manager for Kubernetes
 - [Kustomize](#) - A standalone tool to customize Kubernetes objects through a kustomization file
 - [Artifact Hub](#) - A web-based application designed to facilitate the finding, installing, and publishing of Cloud Native packages and configurations
- Cloud Resource Management
 - [Crossplane](#) - A cloud-native framework for platform engineering that enables users to build their own APIs and services with control planes, extending Kubernetes to manage any resource anywhere
- Developer Workflow Tools
 - [Scaffold](#) - A command line tool that facilitates continuous development for container-based applications
- Platform Extensions

- [kubernetes-fencing](#) - A solution for fencing of stateful application's nodes in Kubernetes
- [KubeVirt](#) - A virtual machine management add-on for Kubernetes
- Operator & Controller Development
 - [Kubebuilder](#) - A framework for building Kubernetes APIs using custom resource definitions (CRDs)
- Resource Optimization
 - [Goldilocks](#) - A utility that can help you identify a starting point for resource requests and limits
- Vendor-specific Tools
 - [eksctl](#) - The official CLI for Amazon EKS
- Batch & Job Scheduling
 - [Kueue](#) - A Kubernetes-native system that manages quotas and resource sharing for batch, HPC, and AI/ML workloads

Cloud-Native Computing

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [Serverless Computing](#) - A cloud computing execution model in which the cloud provider allocates machine resources on demand, taking care of the servers on behalf of their customers

Container as a Service (CaaS)

- Managed Kubernetes
 - [Google Kubernetes Engine \(GKE\)](#) - A managed, production-ready environment for running containerized applications
 - [Amazon Elastic Kubernetes Service](#) - A managed service that makes it easy to run Kubernetes on AWS and on-premises
 - [Azure Kubernetes Service \(AKS\)](#) - A fully managed Kubernetes service for deploying and managing containerized applications
- Simplified Container Hosting
 - [Amazon Elastic Container Service](#) - A fully managed container orchestration service that helps you easily deploy, manage, and scale containerized

applications

- [AWS Fargate](#) - A serverless compute engine for containers that works with both ECS and EKS
- [AWS App Runner](#) - A fully managed service that makes it easy for developers to quickly deploy containerized web applications and APIs, at scale and with no prior infrastructure experience required
- [Azure Container Apps](#) - A fully managed serverless container service built on Kubernetes, integrating KEDA, Dapr, and Envoy for microservices and event-driven workloads
- [Google Cloud Run](#) - A managed compute platform that lets you run containers that are automatically scaled

Function as a Service (FaaS)

- [AWS Lambda](#) - A serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers
- [Azure Functions](#) - An event-driven, serverless compute platform that helps you develop more efficiently using the programming language of your choice
- [Google Cloud Run Functions](#) - A serverless execution environment for building and connecting cloud services

Advanced Runtimes & Isolation

- Sandboxed Runtimes
 - [Kata Containers](#) - An open-source project building a standard implementation of lightweight virtual machines that feel and perform like containers, but provide the workload isolation and security of virtual machines
 - [gVisor](#) - A Linux-compatible sandbox that implements the Linux kernel and its network stack, intercepting system calls to protect the host from containerized applications
 - [libkrun](#) - A dynamic library providing virtualization-based process isolation capabilities
 - [Microsoft eXecution Container \(MXC\)](#) - A cross-platform sandboxed code execution system for running untrusted code with multiple containment backends and policy-driven security controls across Windows, Linux, and macOS
 - [Cloud Hypervisor](#) - An open source Virtual Machine Monitor (VMM) implemented in Rust that focuses on running modern, cloud workloads, with minimal

hardware emulation

- [Firecracker](#) - An open source virtualization technology that is purpose-built for creating and managing secure, multi-tenant container and function-based services
- [AWS Lambda MicroVMs](#) - A serverless compute service that provides VM-level isolation with near-instant startup and resume capabilities, built on Firecracker virtualization technology
- [QEMU microvm](#) - A minimalist machine type without PCI nor ACPI support, designed for short-lived guests, and optimized for both boot time and footprint
- [Docker Sandboxes](#) - The isolated, disposable environments designed to run AI coding agents in lightweight microVMs for enhanced security and system protection
- [Daytona](#) - The secure infrastructure platform for running AI-generated code in isolated sandbox environments with sub-90ms creation times
- [Modal](#) - An AI infrastructure platform that provides a serverless cloud for engineers to build and scale compute-intensive applications in thousands of isolated sandboxes
- Virtualization & Container Storage
 - [virtiofs](#) - A shared file system that lets virtual machines access a directory tree on the host
- Image Services & Distribution
 - [Nydus](#) - A powerful opensource filesystem solution to form a high-efficiency image distribution system for Cloud Native workloads, such as container images, software packages, etc

Cloud-Native Infrastructure

- App Runtimes & Scaling
 - [KEDA \(Kubernetes Event-driven Autoscaling\)](#) - A single-purpose and lightweight component that can be added into any cluster to provide event-driven scale for any container running in the environment
 - [Dapr \(Distributed Application Runtime\)](#) - A portable, event-driven runtime that makes it easy for any developer to build resilient, stateless, and stateful applications that run on the cloud and edge and embraces the diversity of languages and developer frameworks
 - [V8 isolates](#) - An independent instance of the engine with its own heap and its own garbage collector

- Serverless Computing
 - [Knative](#) - A Kubernetes-based platform to build, deploy, and manage modern serverless workloads
- Service Mesh & Discovery
 - [Gateway API](#) - The next generation of Kubernetes Ingress, Load Balancing, and Service Mesh APIs
 - [Istio](#) - An open source service mesh that layers transparently onto existing distributed applications
 - [Kiali](#) - The service mesh observability and configuration tool for Istio
 - [Linkerd](#) - An ultralight, security-first service mesh for Kubernetes
 - [Hashicorp Consul](#) - A service networking solution to connect and secure services across any runtime platform and public or private cloud
- Edge Proxies & Ingress
 - [Envoy Proxy](#) - An open source edge and service proxy
 - [Traefik proxy](#) - A leading modern open source reverse proxy and ingress controller
 - [Contour](#) - A high performance ingress controller for Kubernetes that provides the control plane for the Envoy edge and service proxy
 - [Apache APISIX](#) - An open source API Gateway to help you manage microservices, delivering the ultimate performance, security, and scalable platform for all your APIs and microservices
- Cloud-Native Networking
 - [Project Calico](#) - An open-source project that provides secure network connectivity, network security, and observability for containers, virtual machines, and native host-based workloads
 - [Cilium](#) - An open-source project that provides networking, security, and observability for cloud-native environments
 - [Flannel](#) - A simple and easy way to configure a layer 3 network fabric designed for Kubernetes

CI/CD & GitOps

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > SRE Process

Delivery & Deployment

- Continuous Delivery Tools
 - [Harness](#) - A modern software delivery platform that uses AI and automation to streamline CI/CD, GitOps, and cloud cost management
 - [Jenkins](#) - An open source automation server which enables developers around the world to reliably build, test, and deploy their software
 - [Blue Ocean for Jenkins Pipelines](#) - A project that rethinks the user experience of Jenkins
 - [Python Jenkins](#) - A Python wrapper for the Jenkins REST API
 - [JenkinsPipelineUnit](#) - A testing framework to unit test Jenkins pipelines written in Groovy or Declarative
 - [GitLab CI/CD](#) - A part of GitLab that you can use to automate the builds, integration, and verification of your source code
 - [GitHub Actions](#) - A feature that makes it easy to automate all your software workflows
 - [actionlint](#) - A static checker for GitHub Actions workflow files
 - [act](#) - A tool to run your GitHub Actions locally
 - [Azure Pipelines](#) - A cloud service that you can use to automatically build and test your code project and make it available to other users
- Application Deployment
 - [Kamal](#) - A tool to deploy web apps anywhere

GitOps & Cloud-Native

- GitOps Style CD
 - [ArgoCD](#) - A declarative, GitOps continuous delivery tool for Kubernetes
 - [FluxCD](#) - A tool for keeping Kubernetes clusters in sync with sources of configuration (like Git repositories), and automating updates to configuration when there is new code to deploy
- Cloud-Native Application Delivery
 - [Open Application Model](#) - A specification for describing applications so that they can be deployed and managed across any platform
 - [KubeVela](#) - A modern software delivery platform that makes deploying and operating applications across today's hybrid, multi-cloud environments easier, faster and more reliable

- [Flagger](#) - A progressive delivery tool that automates the release process for applications running on Kubernetes

Integrations & Registries

- Private Package Registries
 - [JFrog Artifactory](#) - A universal DevOps repository manager that allows you to store, manage, and distribute your software artifacts and their dependencies
 - [GitLab Package Registry](#) - A feature that allows you to publish and share packages for a variety of supported package managers
 - [GitHub Packages](#) - A software package hosting service that allows you to host your software packages privately or publicly
 - [Nexus Repository Manager 3](#) - A sophisticated repository manager
 - [Azure Artifacts](#) - A service that enables you to create and share Maven, npm, NuGet, and Python package feeds from public and private sources

System Observability

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > SRE Process

Instrumentation & Platforms

- Concepts
 - [Observability](#) - A measure of how well internal states of a system can be inferred from knowledge of its external outputs
- Instrumentation Libraries
 - [OpenTelemetry](#) - A vendor-neutral open source Observability framework for instrumenting, generating, collecting, and exporting telemetry data such as traces, metrics, and logs
 - [Jaeger](#) - An open source, distributed tracing platform used to monitor and troubleshoot workflows in complex distributed systems
 - [Micrometer](#) - A metrics instrumentation library for JVM-based applications
- Monitoring Tools
 - [Uptime Kuma](#) - An easy-to-use self-hosted monitoring tool
- Managed Platforms

- [Azure Monitor](#) - A comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments
 - [Kusto Query Language](#) - A powerful tool to explore your data and discover patterns, identify anomalies and outliers, create statistical models, and more
 - [App Insights](#) - A feature of Azure Monitor, is an extensible Application Performance Management (APM) service for developers and DevOps professionals
- [AWS CloudWatch](#) - A monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers
- [Datadog](#) - The integrated platform for monitoring & security
- [Sentry](#) - An application monitoring platform that provides error tracking and performance monitoring to help developers see what matters and solve issues faster
- Self-hosted (advanced)
 - [SigNoz](#) - An open-source Datadog or New Relic alternative for logs, metrics, traces, dashboards, alerts, and more
- Visualization Tools
 - [Grafana](#) - The open source data visualization and monitoring solution
 - [Grafonnet](#) - A Jsonnet library for generating Grafana dashboards
 - [gcy](#) - A CLI tool that provides structured access to Grafana resources—dashboards, alerts, SLOs, metrics, logs, and traces—and integrates with AI coding agents for observability-driven development workflows across Grafana Cloud, Enterprise, and self-hosted instances
 - [Kibana](#) - A free and open user interface that lets you visualize your Elasticsearch data and navigate the Elastic Stack

Telemetry Shipment

- Data Shippers
 - [Prometheus exporters](#) - The services that expose Prometheus metrics
 - [node-exporter](#) - An exporter for hardware and OS metrics exposed by *NIX kernels
 - [blackbox-exporter](#) - A tool that allows blackbox probing of endpoints over HTTP, HTTPS, DNS, TCP, ICMP and gRPC
 - [Grafana Alloy](#) - An open source OpenTelemetry collector with built-in Prometheus pipelines and support for metrics, logs, traces, and profiles

- [Fluent Bit](#) - A super fast, lightweight, and highly scalable logging, metrics, and traces processor and forwarder
- [Fluentd](#) - An open source data collector, which lets you unify the data collection and consumption for a better use and understanding of data
- [Filebeat](#) - A lightweight shipper for forwarding and centralizing log data
- [Logstash](#) - An open source server-side data processing pipeline that ingests data from a multitude of sources, transforms it, and then sends it to your favorite "stash"
- [Telegraf](#) - An open source server agent that helps you collect metrics from your stacks, sensors, and systems
- [Metricbeat](#) - A lightweight shipper that you can install on your servers to periodically collect metrics from the operating system and from services running on the server
- [rsyslog](#) - The rocket-fast system for log processing
- Vendor-specific Tools
 - [Azure Monitor Agent](#) - The agent that collects monitoring data from the guest operating system of Azure and hybrid virtual machines
 - [Cloudwatch Agent](#) - The agent you can use to collect both system-level metrics and log files from Amazon EC2 instances and on-premises servers

Telemetry Collection & Storage

- Datastore and Alerting Tools
 - [Prometheus](#) - An open-source systems monitoring and alerting toolkit
 - [PromQL](#) - The Prometheus Query Language
 - [promtool](#) - The command line utility for the Prometheus server
 - [Awesome Prometheus Alerts](#) - A collection of copy-pasteable Prometheus alerting rules spanning over 90 services and exporters
 - [Alertmanager](#) - A tool that handles alerts sent by client applications such as the Prometheus server
 - [amttool](#) - A CLI tool for interacting with the Alertmanager API
 - [InfluxDB](#) - A time series database built from the ground up to handle high write and query loads
 - [InfluxQL](#) - An SQL-like query language for interacting with data in InfluxDB
 - [influx cli](#) - The command line interface for InfluxDB 2.0

- [Grafana Mimir](#) - An open source, horizontally scalable, highly available, multi-tenant, long-term storage for Prometheus
- [Grafana Loki](#) - A horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus
 - [LogQL](#) - The query language for Loki
 - [LogCLI](#) - The command line interface for Loki
- [Grafana Tempo](#) - An open source, easy-to-use and high-scale distributed tracing backend
 - [TraceQL](#) - A query language designed for selecting traces
- [ElasticSearch](#) - An open source distributed, RESTful search and analytics engine, scalable data store, and vector database
 - [Elastic Common Schema](#) - An open source specification, developed with support from the Elastic user community
 - [Ingest pipelines](#) - A feature that lets you perform common transformations on your data before indexing
 - [Dissect and Grok](#) - The processors that let you extract structured fields out of a single text field
- [Gaphite](#) - A highly scalable real-time graphing system
- [Grafana Alerting](#) - A feature that allows you to create and manage alerts for your data
- [OpenObserve](#) - An open-source observability platform designed for modern applications

SRE (Site Reliability Engineering)

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > SRE Process
- [Site Reliability Engineering](#) - A discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems
 - [Service Level Objectives \(SLOs\)](#) - A target value or range of values for a service level that is measured by a service level indicator (SLI)
 - [Dickerson's Hierarchy of Service Reliability](#) - A model that illustrates the foundational elements required to build and maintain reliable services, often visualized as a pyramid
 - [The Four Golden Signals](#) - The four key metrics (Latency, Traffic, Errors, and

Saturation) that Google SREs use for monitoring user-facing systems

- [Ishikawa diagram](#) - A causal diagram created by Kaoru Ishikawa that shows the potential causes of a specific event

Fleet Management & Operations

- Fleet Management
 - [AWS Systems Manager](#) - A secure end-to-end management solution for resources on AWS and in multicloud and hybrid environments
 - [Azure Automation](#) - A cloud-based automation and configuration service that supports consistent management across your Azure and non-Azure environments
 - [Azure Update Manager](#) - A unified service to help manage and govern updates for all your machines
- Backup
 - Vendor-specific Tools
 - [AWS Backup](#) - A fully managed service that centralizes and automates data protection across AWS services, in the cloud, and on premises
 - [Azure Backup](#) - A service that provides simple, secure, and cost-effective solutions to back up your data and recover it from the Microsoft Azure cloud
 - K8s-specific Tools
 - [Velero](#) - An open source tool to safely back up and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes
 - Generic
 - [Barman](#) - A disaster recovery solution for PostgreSQL databases, designed to ensure business continuity by simplifying online hot backups
 - [Restic](#) - A fast, secure, efficient backup program
- Runbook Automation
 - [RunDeck](#) - An open source automation platform that helps you automate routine operational procedures in data center or cloud environments
- AIOps & Autonomous Agents
 - [Azure SRE Agent](#) - An AI-powered service designed to automate Site Reliability Engineering practices by monitoring, diagnosing, and helping resolve incidents
 - [Mezmo Aura](#) - An open-source agentic harness designed specifically for Site Reliability Engineering (SRE) and production AI operations

Chaos Engineering

- Concepts
 - [Chaos Engineering](#) - The practice of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production
 - [Principles of chaos engineering](#) - The principles that define the practice of chaos engineering
- Chaos Engineering Tools
 - [Litmus](#) - A cloud-native chaos engineering framework for Kubernetes
 - [Chaos Mesh](#) - A cloud-native Chaos Engineering platform that orchestrates chaos on Kubernetes environments
 - [Toxiproxy](#) - A TCP proxy to simulate network and system conditions for chaos and resiliency testing
 - [kubenvaders](#) - A gamified chaos engineering tool for Kubernetes

FinOps

- Concepts
 - [FinOps principles](#) - The cultural practice of bringing financial accountability to the variable spend model of cloud
- FinOps Tools
 - [FinOps toolkit](#) - A collection of tools, resources, and best practices for implementing FinOps in your organization
 - [AWS Cost Explorer](#) - A tool that enables you to view and analyze your costs and usage
 - [Infracost](#) - A tool that shows cloud cost estimates for Terraform, CloudFormation, and other infrastructure-as-code projects
 - [OpenCost](#) - The open source solution for monitoring Kubernetes spend
 - [Cloud Custodian](#) - A rules engine for managing public cloud accounts and resources

Performance & Load Testing

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > SRE Process

- [Performance Testing](#) - The practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload

Performance Testing Tools

- [Hyperfine](#) - A command-line benchmarking tool
- [Locust](#) - An easy-to-use, distributed, user load testing tool
- [Grafana k6](#) - The open-source load testing tool that makes performance testing easy and productive for engineering teams
- [Gatling](#) - The load testing tool for programmers that helps engineering teams shift performance concerns left
- [Apache Jmeter](#) - A pure Java application designed to load test functional behavior and measure performance
- [ab](#) - A tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server
- [stress-ng](#) - A tool that imposes configurable amounts of CPU, memory, I/O, and disk stress on the system
- [sysbench](#) - A scriptable multi-threaded benchmark tool based on LuaJIT
- [fio](#) - A tool that will spawn a number of threads or processes doing a particular type of I/O action as specified by the user
- [iPerf](#) - The ultimate speed test tool for TCP, UDP and SCTP
- [Speedtest CLI](#) - A command line interface that measures internet connection performance metrics like download, upload, latency and packet loss natively without relying on a web browser
- [plow](#) - An HTTP(S) benchmarking tool written in Golang that conducts load testing with concurrent connections while displaying real-time performance metrics through both a web UI and terminal interface
- [loadgen-rs](#) - A h2load-compatible HTTP benchmark client written in Rust, supporting HTTP/1.1, HTTP/2, and HTTP/3 (QUIC) with commandline mode and distributed mode

04 - Security & Privacy

Security Foundations

Relevant DSS-P Skills

- 4. Security > 4.1 Security Management > Security System Construction and Operation

- 4. Security > 4.1 Security Management > Security Management

- **Information security** - The practice of protecting information by mitigating information risks
- **Vulnerability** - A weakness which can be exploited by a threat actor
- **Threat** - A potential negative action or event facilitated by a vulnerability
- **Shared Responsibility Model** - A security and compliance framework that outlines the responsibilities of cloud service providers (CSPs) and customers for securing every aspect of the cloud environment

Common Threats & Attack Vectors

- **Malware** - Any software intentionally designed to cause disruption to a computer, server, client, or computer network
- **Ransomware** - A type of malware from cryptovirology that threatens to publish the victim's personal data or perpetually block access to it unless a ransom is paid
- **Social engineering** - The psychological manipulation of people into performing actions or divulging confidential information
- **Phishing** - A type of social engineering where an attacker sends a fraudulent message designed to trick a person into revealing sensitive information
- **Business Email Compromise (BEC)** - A type of phishing attack in which an attacker impersonates a high-level executive and attempts to trick an employee or customer into transferring money or sensitive data
- **Infostealer** - A type of Trojan horse designed to gather information from a system
- **Mirai (malware)** - A malware that turns networked devices running Linux into remotely controlled bots that can be used as part of a botnet in large-scale network attacks
- **Think before you Click(Fix)** - A social engineering technique that tricks users into running malicious commands on their devices by taking advantage of their target's tendency to solve minor technical issues
- **Evilginx** - A man-in-the-middle attack framework used for phishing login credentials along with session cookies

Modern Security Architectures

- **Zero trust security model** - An approach to the design and implementation of IT systems where trust is never granted implicitly and verification is required for everyone

Security Training & Competitions

- [Capture the flag \(cybersecurity\)](#) - A hacking contest where participants use specialized knowledge and techniques to find hidden "Flags" (answers) and compete for the highest total score
- Platforms
 - [CTFd](#) - The easiest Capture The Flag platform to host your own cyber security workshop, providing a rock solid base that is easily customizable with themes and plugins

Cryptography & Data Protection

Relevant DSS-P Skills

- 4. Security > 4.2 Security Technology > Secure Design, Development, and Construction

Core Cryptography

Hashing

- Hash Function
 - [MD5](#) - A cryptographically broken but still widely used hash function producing a 128-bit hash value
 - [SHA-2 \(SHA-224, SHA-256, SHA-384, SHA-512\)](#) - A set of cryptographic hash functions designed by the United States National Security Agency (NSA)
 - [Bcrypt](#) - A password-hashing function based on the Blowfish cipher
 - [Scrypt](#) - A password-based key derivation function created by Colin Percival

Symmetric-key Cryptography

- [Symmetric-key algorithm](#) - Algorithms for cryptography that use the same cryptographic keys for both the encryption of plaintext and the decryption of ciphertext
- Block Cipher
 - [AES](#) - A specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001
- Stream Cipher
 - [Salsa20 ChaCha](#) - A variant of Salsa20 that increases the diffusion per round

while achieving the same or slightly better performance

- MAC (Message Authentication Code)
 - [HMAC](#) - A specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key
- Modes of Operation
 - [CBC \(Cipher block chaining\)](#) - A mode of operation for a block cipher where a block of plaintext is XORed with the previous ciphertext block before being encrypted
 - [GCM \(Galois/Counter Mode\)](#) - A mode of operation for symmetric-key cryptographic block ciphers which is widely adopted for its performance
 - [CCM](#) - A mode of operation for cryptographic block ciphers designed to provide both authentication and confidentiality

Public-key Cryptography

- [Digital signature](#) - A mathematical scheme for verifying the authenticity of digital messages or documents
- [Public-key cryptography](#) - A cryptographic system that uses pairs of keys
 - [RSA](#) - A public-key cryptosystem that is widely used for secure data transmission
 - [EdDSA](#) - A digital signature scheme using a variant of Schnorr signature based on twisted Edwards curves
- Key Agreement
 - [Diffie-Hellman key exchange](#) - A method of securely exchanging cryptographic keys over a public channel
 - [Elliptic-curve Diffie-Hellman](#) - A key agreement protocol that allows two parties to establish a shared secret over an insecure channel
- Encryption Schemes
 - [RSAES-PKCS1-v1_5](#) - An older Encryption/decryption Scheme (ES) was first standardized in version 1.5 of PKCS #1 and is known to be vulnerable
 - [RSAES-OAEP](#) - A padding scheme that enhances RSA encryption by adding randomness and preventing partial decryption, it was standardized in PKCS#1 v2 and RFC 2437
- Signature Schemes
 - [RSASSA-PKCS1-v1_5](#) - A Signature Scheme with Appendix (SSA) was first standardized in version 1.5 of PKCS #1 and is considered unforgeable according to Jager et al. (2018)

- [DSA](#) - A public-key cryptosystem and Federal Information Processing Standard for digital signatures, it is based on the mathematical concept of modular exponentiation and the discrete logarithm problem
- [ECDSA](#) - A variant of the Digital Signature Algorithm (DSA) that utilizes elliptic-curve cryptography
- Key formats
 - [PKCS #1: RSA Cryptography Specifications](#) - A standard that provides the basic definitions of and recommendations for implementing the RSA algorithm for public-key cryptography
 - [PKCS #12: Personal Information Exchange Syntax](#) - A file format for storing multiple cryptographic objects in a single file
- Cryptographic Standards & Formats
 - [Cryptographic Message Syntax](#) - The IETF's standard for cryptographically protected messages, used by cryptographic schemes and protocols to digitally sign, digest, authenticate, or encrypt any form of digital data

Public Key Infrastructure (PKI)

- [Public Key Infrastructure \(PKI\)](#) - A set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates
- [Certificate authority \(CA\)](#) - An entity that stores, signs, and issues digital certificates
- Registration authority (RA)
- Validation authority (VA)
- Protocols & Standards
 - [Simple Certificate Enrollment Protocol](#) - A protocol for enrolling X.509 certificates in a secure and automated manner
- Validation & Enrollment
 - [Domain Control Validation](#) - A process used by certificate authorities (CAs) to verify that the person or organization requesting a certificate has control over the domain(s) listed in the certificate
- Trust Stores
 - [Certifi](#) - A carefully curated collection of Root Certificates for validating the trustworthiness of SSL certificates while verifying the identity of TLS hosts
- [Public key certificate](#) - An electronic document used to prove the validity of a public key

- Domain Validated (DV)
- Organization Validated (OV)
- Extended Validation (EV)
- [Let's Encrypt](#) - A nonprofit Certificate Authority providing TLS certificates
 - [certbot](#) - A free, open source software tool for automatically using Let's Encrypt certificates on manually-administrated websites to enable HTTPS
 - [lego](#) - A Let's Encrypt client and ACME library written in Go
- [ACME \(Automatic Certificate Management Environment\)](#) - A communications protocol for automating interactions between certificate authorities and their users' web servers
- [mkcert.org](#) - A simple tool for making locally-trusted development certificates
- [cert-manager](#) - A powerful and extensible X.509 certificate controller for Kubernetes and OpenShift
- [cfssl](#) - Cloudflare's PKI toolkit

Secrets Management

- [Vault](#) - A tool for securely accessing secrets like API keys, passwords, or certificates
- [OneCLI](#) - An open-source credential vault and proxy gateway for AI agents that stores secrets in an encrypted vault and injects them into agent requests without exposing keys
- [SOPS](#) - An editor of encrypted files that supports YAML, JSON, ENV, INI and BINARY formats
- [git-secret](#) - A bash tool to store your private data inside a git repo
- Kubernetes Ecosystem
 - [Sealed Secrets](#) - A Kubernetes controller and tool for one-way encrypted Secrets
 - [Secrets Store CSI Driver](#) - A driver that allows Kubernetes to mount multiple secrets, keys, and certs stored in enterprise-grade external secrets stores into pods as a volume
 - [External Secrets Operator](#) - A Kubernetes operator that integrates external secret management systems like AWS Secrets Manager, HashiCorp Vault, Google Secrets Manager, Azure Key Vault, and IBM Cloud Secrets Manager
- Vendor Services
 - [Azure Key Vault](#) - A cloud service to safeguard cryptographic keys and other secrets used by cloud apps and services

- [Google Cloud Secret Manager](#) - A secure and convenient storage system for API keys, passwords, certificates, and other sensitive data
- [AWS Key Management Service](#) - A service that makes it easy for you to create and manage cryptographic keys
- [AWS Secrets Manager](#) - A secrets management service that helps you protect access to your applications, services, and IT resources

Applied Cryptography & Tools

- Advanced Cryptography Topics
 - [Post-quantum cryptography](#) - Cryptographic algorithms that are thought to be secure against a cryptanalytic attack by a quantum computer
 - Information Hiding
 - [Steganography](#) - The practice of concealing a file, message, image, or video within another file, message, image, or video
 - [Digital watermarking](#) - A kind of marker covertly embedded in a noise-tolerant signal such as an audio, video or image data
- End-to-end Encryption Tools
 - [age](#) - A simple, modern and secure file encryption tool, format, and Go library
 - [Pretty Good Privacy \(PGP\)](#) - A data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication
 - [OpenPGP](#) - A non-proprietary protocol for exchanging public keys and encrypted messages
 - [keys.openpgp.org](#) - A public key server for OpenPGP
 - [GnuPG](#) - A free software replacement for the PGP cryptographic software suite
 - [Gpg4win](#) - A Windows software package that facilitates the secure transport of emails and files with the help of OpenPGP
- Cryptographic Libraries
 - [PyCryptodome](#) - A self-contained Python package of cryptographic primitives
 - [Python cryptography](#) - A package designed to expose cryptographic primitives and recipes to Python developers
 - [Go Cryptography](#) - A collection of Go cryptography libraries
 - [Botan](#) - A cryptography library written in C++

Identity & Access Management (IAM)

Relevant DSS-P Skills

- 4. Security > 4.2 Security Technology > Secure Design, Development, and Construction
- 2. Data Preparation & Utilization > 2.3 Data Management > Improvement of Data Quality and Safety

Integrated IAM

- [Identity management](#) - A framework of policies and technologies for ensuring that the proper people in an enterprise have the appropriate access to technology resources
- Self-hosted IAM Platforms
 - [FusionAuth CE](#) - The self-hosted, community supported version of FusionAuth
 - [KeyCloak](#) - An open source identity and access management solution
 - [FreeIPA](#) - An integrated security information management solution combining Linux, 389 Directory Server, MIT Kerberos, NTP, DNS, and a certificate system
- Cloud IAM Services
 - [Microsoft Entra ID](#) - A cloud-based identity and access management service
 - [AWS IAM](#) - A service that helps you securely control access to AWS resources
 - [Amazon Cognito](#) - A service that lets you add user sign-up, sign-in, and access control to your web and mobile apps
 - [Auth0](#) - A flexible, drop-in solution to add authentication and authorization services to your applications
- [Directory service](#) - A service that maps the names of network resources to their respective network addresses
 - [LDAP](#) - An open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services
 - [OpenLDAP](#) - An open source implementation of the Lightweight Directory Access Protocol
 - [389 Directory Server](#) - A free and open source software project developed by Red Hat for Linux systems
- Specifications
 - [Decentralized Identifiers \(DIDs\)](#) - A new type of identifier that enables verifiable,

decentralized digital identity

- [System for Cross-domain Identity Management \(SCIM\)](#) - A specification designed to make managing user identities in cloud based applications and services easier

Authentication (AuthN)

- [Authentication](#) - The act of proving an assertion, such as the identity of a computer system user
- [Mutual authentication](#) - A process in which both parties in a communications link authenticate each other
- [Multi-factor authentication \(MFA\)](#) - A method that requires multiple verification methods for access
- [3-D Secure](#) - A security protocol designed to be an additional security layer for online credit and debit card transactions
- [Single sign-on \(SSO\)](#) - A service that allows one login for multiple applications
- Protocols & Standards
 - [OpenID Connect](#) - A simple identity layer on top of the OAuth 2.0 protocol
 - [SAML](#) - A standard for logging users into applications
 - [WS-Federation](#) - A specification that defines mechanisms used to broker trust and manage identities, attributes and authentication between participating Web services
 - [FIDO2](#) (WebAuthn, CTAP, Passkeys) - A set of specifications that enables users to leverage common devices to easily and securely authenticate to online services
 - Relying party - The website or online service that wants to verify a user's identity (e.g., your bank's website)
 - Authenticator - The device or software that securely stores cryptographic keys and performs authentication for the user
 - Client - The software on the user's device, typically a web browser or operating system component, that communicates between the Relying Party and the Authenticator
 - [WebAuthn](#) - An API for accessing Public Key Credentials
 - [CTAP](#) - A protocol that enables an external authenticator to communicate with a client platform
 - [Passkeys](#) - A phishing-resistant replacement for passwords
 - [SPIFFE](#) - The Secure Production Identity Framework for Everyone

- [Kerberos](#) - A computer network authentication protocol that works on the basis of tickets
- [SSPI \(Security Support Provider Interface\)](#) - A Win32 API that allows an application to use various security models available on a computer or network without changing the interface to the security system
- Credentials & Tokens
 - [Basic authentication](#) - A method for an HTTP user agent to provide a username and password when making a request
 - [JSON Web Token \(JWT\)](#) - An Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims
 - [nodejs jsonwebtoken](#) - An implementation of JSON Web Tokens for Node.js
 - [TOTP \(Time-Based One-Time Password\)](#) - A temporary passcode, generated by an algorithm, for use in authenticating access to computer systems
 - [AWS Signature Version 4 \(SigV4\)](#) - The process to add authentication information to AWS API requests sent by HTTP
- Platforms & Tools
 - [Dex](#) - A Federated OpenID Connect Provider
 - [Firebase Authentication](#) - A service that provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app
 - [Supabase Auth](#) - A service that provides user management and access control for Supabase projects
 - [ReCAPTCHA](#) - A CAPTCHA system that enables web hosts to distinguish between human and automated access to websites
 - [Microsoft Authentication Library \(MSAL\)](#) - A library that helps developers integrate authentication and authorization into applications
 - [Application Default Credentials \(ADC\)](#) - A mechanism used by Google Cloud client libraries to automatically find credentials based on the application environment
 - [Limen](#) - A lightweight, composable authentication and authorization library for Go that provides sessions, password hashing, OAuth, and CSRF protection

Authorization (AuthZ)

- [Authorization](#) - The function of specifying access rights/privileges to resources
- Access Control Models

- [Access control list \(ACL\)](#) - A list of permissions associated with a system resource
- [Attribute-based access control \(ABAC\)](#) - A model grants access based on user attributes
- [Discretionary access control \(DAC\)](#) - A model allows users to control access to their own resources
- [Mandatory access control \(MAC\)](#) - A model enforces access policies based on security labels
- [Role-based access control \(RBAC\)](#) - A policy-neutral access-control mechanism defined around roles and privileges
 - [Azure RBAC](#) - A system that enables fine-grained access management of Azure resources
 - Security principal kinds: User, Group, Service Principal, Managed Identity
 - [Entra ID RBAC](#) - A system that provides fine-grained access management of Microsoft Entra resources
- Protocols & Standards
 - [OAuth 2.0 Authorization Framework](#) - An open standard for access delegation
 - Resource owner - the user who owns the data or resources that are being accessed
 - Resource server - the server that hosts the protected resources
 - Client - an application or service that wants to access the resources on behalf of the resource owner
 - Authorization server - the server that issues access tokens to the client
- Platforms & Tools
 - [Permify](#) - An open-source authorization service that helps you to create any kind of authorization system with its Golang API
 - [Azure Shared Access Signature \(SAS\)](#) - A signed URI that points to one or more storage resources and includes a token that specifies the permissions and interval of access

Secure Development Lifecycle (DevSecOps)

Relevant DSS-P Skills

- 4. Security > 4.2 Security Technology > Secure Design, Development, and Construction

Secure Design & Modeling

- [Threat modeling](#) - A process by which potential threats can be identified, enumerated, and prioritized from a hypothetical attacker's point of view
- [OWASP Threat Modeling](#) - Guidance from the OWASP Security Culture project on integrating threat modeling into the development lifecycle
- [STRIDE model](#) - A mnemonic for categorizing computer security threats into six categories
- [MITRE ATT&CK](#) - A globally-accessible knowledge base of adversary tactics and techniques
 - Tactics: The high-level objectives or goals that an adversary aims to achieve during an attack
 - Techniques: The specific methods or ways adversaries achieve their tactical objectives
 - Procedures: The specific implementations or variations of techniques that adversaries utilize in their operations
- Modeling Tools
 - [OWASP Threat Dragon](#) - A free, open-source, cross-platform threat modeling application
 - [threatspec](#) - A tool that allows you to define threat models as code

Secure Development Practices

- [Secure Software Development Framework \(SSDF\)](#) - A set of fundamental, sound, and secure software development practices
- [Microsoft Security Development Lifecycle \(SDL\)](#) - A software development process that helps developers build more secure software and address security compliance requirements while reducing development cost
- [OWASP Application Security Verification Standard \(ASVS\)](#) - A standard for performing application-level security verifications
- [OWASP Security Champions](#) - A program to embed security expertise and culture within development teams
- [OWASP Cheat Sheet Series](#) - A collection of concise cheat sheets on various security topics
- [OWASP LLM Top 10](#) - A guide to the top 10 risks, vulnerabilities, and mitigations for developing and securing generative AI and large language model applications across the development, deployment, and management lifecycle

- Coding Standards
 - [MISRA C](#) - A set of guidelines for the use of the C programming language in safety-critical embedded systems, originally developed for the automotive industry
 - [CERT Secure Coding Standards](#) - A collection of programming security guidelines for C, C++, Java, Perl, and other languages published by Carnegie Mellon's CERT

Web Application Security

- Security Mechanisms & Policies
 - [SOP \(Same-origin policy\)](#) - An important concept in the web application security model
 - [CORS \(Cross-Origin Resource Sharing\)](#) - A mechanism that allows restricted resources on a web page to be requested from another domain
 - [CSP \(Content Security Policy\)](#) - An added layer of security that helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks
 - [HSTS \(HTTP Strict Transport Security\)](#) - A web security policy mechanism that helps to protect websites against protocol downgrade attacks and cookie hijacking
 - [Cross-origin isolation](#) - A web security feature that allows a web page to use powerful features like `SharedArrayBuffer` and `performance.measureUserAgentSpecificMemory()`
- Common Vulnerabilities & Attacks
 - [Cross-site request forgery \(CSRF\)](#) - A type of malicious exploit of a website where unauthorized commands are submitted from a user that the web application trusts
 - [Cross-site scripting \(XSS\)](#) - A type of security vulnerability typically found in web applications
 - [DNS rebinding](#) - A type of attack that allows a malicious web page to bypass the same-origin policy by exploiting the Domain Name System
 - [SSRF \(Server-side request forgery\)](#) - A type of exploit where an attacker can abuse the functionality on the server to read or modify internal resources
- Privacy & Transparency
 - [Privacy sandbox](#) - Google's initiative to create web technologies that protect people's privacy online and give companies and developers the tools to build thriving digital businesses

- [security.txt](#) - A proposed standard which allows websites to define security policies for researchers

Application Security Testing (AST)

- Integrated Security Platforms
 - [GitHub Advanced Security](#) - A suite combining Secret Protection to stop leaks before they start and Code Security to fix vulnerabilities in your code, delivering security that moves at the speed of development
- Static Analysis (SAST)
 - [SonarQube Server](#) - A self-managed, automatic code review tool that systematically helps you deliver clean code
 - [GitLab SAST](#) - A tool that checks your source code for known vulnerabilities
 - [Bandit](#) (for Python) - A tool designed to find common security issues in Python code
 - [Semgrep OSS](#) - A fast, open-source, static analysis tool for finding bugs and enforcing code standards
 - [Fluid attacks](#) - A security tool that allows you to find vulnerabilities in your source code, containers and dependencies
 - [CodeQL](#) - A semantic code analysis engine that lets you query code as though it were data to find vulnerabilities and their variants
- Dynamic Analysis (DAST)
 - [ZAP](#) - The world's most widely used web app scanner, free and open source, and a community based GitHub Top 1000 project that anyone can contribute to
 - [Nuclei](#) - A fast and customizable vulnerability scanner powered by the global security community and built on a simple YAML-based DSL
 - [sqlmap](#) - An open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers
- Secret Detection
 - [GitLab Secret Detection](#) - A tool that scans your repository's history for secrets
 - [Gitleaks](#) - A SAST tool for detecting and preventing hardcoded secrets like passwords, api keys, and tokens in git repos
 - [secretlint](#) - A pluggable linting tool to prevent committing credentials
 - [Talisman](#) - A tool that installs a hook to your repository to ensure that potential secrets or sensitive information do not get committed

- [TruffleHog](#) - A tool that scans your environment for secrets, digging deep into commit history and branches
- [Whispers](#) - A static code analysis tool designed for parsing various common data formats in search of hardcoded credentials
- AI-orchestrated Penetration Testing
 - [PentestGPT](#) - An automated penetration testing framework powered by Large Language Models (LLMs)
 - [PentAGI](#) - An open-source, fully autonomous AI agent system designed for automated security testing
 - [Strix](#) - A set of autonomous AI agents that act like real hackers to run code dynamically, find vulnerabilities, and validate them through actual proof-of-concepts
 - [CAI](#) - A lightweight, open-source framework that empowers security professionals to build and deploy AI-powered offensive and defensive automation
 - [HexStrike AI](#) - An advanced MCP server that lets AI agents autonomously run 150+ cybersecurity tools for automated pentesting, vulnerability discovery, bug bounty automation, and security research
 - [Zen AI Pentest](#) - An autonomous, AI-powered penetration testing framework that combines cutting-edge language models with professional security tools
 - [Project Glasswing](#) - An initiative providing access to Anthropic's Claude Mythos Preview model for defensive cybersecurity, capable of autonomously finding vulnerabilities in critical software infrastructure

Infrastructure as Code (IaC) Security

- [Trivy](#) - A comprehensive and versatile security scanner
- [Defender for Cloud CLI](#) - A developer-first command-line tool that orchestrates security scans in CI/CD pipelines and uploads results to Microsoft Defender for Cloud for posture management and prioritization
- [checkov](#) - A static code analysis tool for scanning infrastructure as code (IaC) files for misconfigurations
- [Haskell Dockerfile Linter](#) - A smarter Dockerfile linter that helps you build best practice Docker images
- [kube-score](#) - A tool that performs static code analysis of your Kubernetes object definitions
- [kubesecc](#) - A security risk analysis for Kubernetes resources

- [PSRule](#) - A cross-platform PowerShell module with commands to test and verify infrastructure as code (IaC)
 - [PSRule for Azure](#) - A suite of rules to validate Azure resources and infrastructure as code (IaC) using PSRule
- [ComplianceAsCode](#) - A project that provides security automation content in various formats like SCAP, Bash, and Ansible
- [complyctl](#) - A command-line tool that uses OSCAL to streamline compliance assessment activities

Software Supply Chain Security (SSCS)

- Composition Analysis (SCA)
 - SBOM Generation
 - [Syft](#) - A CLI tool and Go library for generating a Software Bill of Materials (SBOM) from container images and filesystems
 - [OWASP CycloneDX format](#) - A lightweight Software Bill of Materials (SBOM) standard designed for use in application security contexts
 - [SPDX format](#) - An open standard for communicating Software Bill of Materials (SBOM) information
 - Vulnerability Scanning
 - [Grype](#) - A vulnerability scanner for container images and filesystems
 - [OSV-scanner](#) - The official vulnerability scanner for OSV
 - [Safety](#) - A tool that checks your installed dependencies for known security vulnerabilities
 - [Clair](#) - An open-source project for the static analysis of vulnerabilities in application containers
 - [GitLab Container Scanning](#) - A tool that checks your Docker images for known vulnerabilities
 - [JFrog Xray](#) - An application security tool that identifies vulnerabilities and license compliance issues across the software supply chain
 - License & Dependency Analysis
 - [Feluda](#) - A blazing fast dependency graph generator for Python projects
 - Automated Dependency Updates
 - [Dependabot Core](#) - The library at the heart of Dependabot security / version updates

- Frameworks & Assessment
 - [SLSA framework](#) - A security framework of standards and controls to prevent tampering, improve integrity, and secure packages and infrastructure
 - [in-toto](#) - A framework to protect software supply chain integrity
 - [OpenSSF Scorecard](#) - An automated tool that assesses a number of important heuristics ("checks") associated with software security and assigns each check a score of 0-10
- Provenance & Artifact Metadata
 - [GUAC](#) - An open source tool that aggregates software security metadata into a high fidelity graph database
- Secure Distribution & Updates
 - [The Update Framework \(TUF\)](#) - A framework for securing software update systems, providing protection even against attackers that compromise the repository or signing keys
- Code Signing & Integrity
 - [Sigstore](#) (Fulcio, Rekor, Cosign) - A new standard for signing, verifying and protecting software
- Notable Attacks
 - [Shai-Hulud npm Supply Chain Attack](#) - A self-propagating worm that harvests sensitive data via post-install scripts and automatically publishes malicious versions of accessible packages to spread further

Runtime & Operational Security

Relevant DSS-P Skills

- 4. Security > 4.1 Security Management > Incident Response and Business Continuity
- 4. Security > 4.2 Security Technology > Security Operation, Maintenance, and Monitoring

Cloud-Native Application Protection (CNAPP)

- [The 4 Cs of Cloud-Native Systems](#) - A defense-in-depth approach that divides security strategies into four distinct layers to provide multilayered protection for cloud-native applications
- [Microsoft Defender for Cloud](#) - A cloud-native application protection platform

(CNAPP) with a set of security measures and practices designed to protect cloud-based applications

- Cloud Security Posture Management (CSPM)
 - [AWS Security Hub](#) - A cloud security posture management (CSPM) service that performs security best practice checks, aggregates alerts, and enables automated remediation
 - [cnquery](#) - A cloud-native, graph-based security tool that allows you to query your entire infrastructure as data
- Cloud Workload Protection Platform (CWPP)
 - [Amazon Inspector](#) - An automated security assessment service that helps improve the security and compliance of applications deployed on AWS
 - [Falco](#) - The cloud-native runtime security project
 - [Tracee](#) - A powerful runtime security and forensics tool for Linux
 - [ClamAV](#) - An open source antivirus engine for detecting trojans, viruses, malware & other malicious threats
 - [YARA](#) - The pattern matching swiss knife for malware researchers

Security Operations & Monitoring (SecOps)

- Detection & Response
 - [Endpoint detection and response \(EDR\)](#) - A cybersecurity technology that addresses the need for continuous monitoring and response to advanced threats
 - [Extended detection and response \(XDR\)](#) - A SaaS-based, vendor-specific, security threat detection and incident response tool
 - [Managed detection and response \(MDR\)](#) - An outsourced service that provides organizations with threat hunting services and responds to threats once they are discovered
- SIEM & SOAR
 - [Security orchestration, automation and response \(SOAR\)](#) - A stack of compatible software programs that allow an organization to collect data about security threats
 - [Microsoft Sentinel](#) - A scalable, cloud-native, security information and event management (SIEM) and security orchestration, automation, and response (SOAR) solution
 - [Amazon GuardDuty](#) - A threat detection service that continuously monitors for

malicious activity and unauthorized behavior

- Detections & Auditing
 - [Sigma Detection Format](#) - A generic and open signature format that allows you to describe relevant log events in a straightforward manner
 - [AWS CloudTrail](#) - An AWS service that helps you enable operational and risk auditing, governance, and compliance of your AWS account
 - [AWS Config](#) - A service that enables you to assess, audit, and evaluate the configurations of your AWS resources

Policy Enforcement

- [Open Policy Agent \(OPA\)](#) - An open source, general-purpose policy engine that unifies policy enforcement across the stack
 - [Rego](#) - A high-level declarative language used to write policies for OPA
 - [Conftest](#) - A utility to help you write tests against structured configuration data
- Cloud Policy Engines
 - [Azure Policy](#) - A service to achieve real-time cloud compliance at scale with consistent resource governance
- Kubernetes Policy Engines
 - [Gatekeeper](#) - A customizable validating webhook that enforces policies executed by the Open Policy Agent (OPA)
 - [Kyverno](#) - A policy engine designed for Kubernetes

Digital Forensics & Incident Response (DFIR)

- Concepts
 - [Computer security incident management](#) - The monitoring and detection of security events on a computer or computer network, and the execution of proper responses to those events
 - [Digital forensics](#) - A branch of forensic science that involves the recovery, investigation, examination, and analysis of material found in digital devices, often in relation to mobile devices and computer crime
 - [Computer forensics](#) - A branch of digital forensic science pertaining to evidence found in computers and digital storage media
- Tools & Platforms
 - [Volatility](#) - The world's most widely used framework for extracting digital

artifacts from volatile memory (RAM) samples

- [Autopsy](#) - A digital forensics platform and graphical interface to The Sleuth Kit and other digital forensics tools

Secure Communications & Networking

Relevant DSS-P Skills

- 4. Security > 4.2 Security Technology > Secure Design, Development, and Construction

Transport Layer Security (TLS)

- [Transport Layer Security \(TLS\)](#) - A cryptographic protocol designed to provide communications security over a computer network
- [Server Name Indication \(SNI\)](#) - An extension to the Transport Layer Security (TLS) computer networking protocol
- Tools & Libraries
 - [testssl.sh](#) - A free command line tool which checks a server's service on any port for the support of TLS/SSL ciphers and protocols
 - [OpenSSL library](#) - A software library for applications that secure communications over computer networks against eavesdropping
 - [stunnel](#) - A proxy designed to add TLS encryption functionality to existing clients and servers
 - [Squid SSL Bump](#) - A feature of Squid proxy that allows it to intercept, decrypt, and re-encrypt SSL/TLS traffic
- Vulnerabilities
 - [Lucky Thirteen attack](#) - A timing attack against the TLS protocol that allows an attacker to decrypt ciphertext

Secure Shell (SSH)

- [Secure Shell \(SSH\)](#) - A cryptographic network protocol for operating network services securely over an unsecured network
- Tools & Libraries
 - [OpenSSH](#) - The premier connectivity tool for remote login with the SSH protocol
 - [PuTTY](#) - A free implementation of SSH and Telnet for Windows and Unix platforms

- [ssh-audit](#) - A tool for SSH server & client configuration auditing
- [keychain](#) - A manager for ssh-agent, typically started from ~/.bash_profile

Firewalls & Network Protection

- Web Application Firewall (WAF)
 - [AWS WAF](#) - A web application firewall that helps protect your web applications or APIs against common web exploits and bots
 - [Azure Web Application Firewall](#) - A cloud-native service that protects web apps from common web-hacking techniques and vulnerabilities
- Network-level Protection
 - [AWS Shield](#) - A managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS
 - [Azure DDoS Protection](#) - A service that provides countermeasures against the most sophisticated DDoS threats
 - [Fail2ban](#) - An intrusion prevention software framework that protects computer servers from brute-force attacks
 - [Snort](#) (IPS) - The foremost Open Source Intrusion Prevention System (IPS) in the world
- Host-based Firewalls
 - [netfilter](#) (iptables, nftables) - A framework inside the Linux kernel that enables packet filtering, network address translation, and other packet mangling
 - [Uncomplicated Firewall \(ufw\)](#) - A program for managing a netfilter firewall

Email & DNS Security

- Email Security
 - [STARTTLS](#) - A mechanism allows plain text communication protocols to upgrade to an encrypted connection
 - [SASL \(Simple Authentication and Security Layer\)](#) - A framework for authentication and data security in Internet protocols
 - [SPF \(Sender Policy Framework\)](#) - An email authentication method designed to detect forging sender addresses during the delivery of the email
 - [DKIM \(DomainKeys Identified Mail\)](#) - An email authentication method designed to detect forged sender addresses in email
 - [DMARC \(Domain-based Message Authentication, Reporting & Conformance\)](#) - An

email authentication, policy, and reporting protocol

- [S/MIME](#) - A standard provides cryptographic security services like authentication, message integrity, non-repudiation, privacy, and data security for electronic messaging applications
- DNS Security
 - [DNSSEC](#) - A feature of the Domain Name System (DNS) that authenticates responses to domain name lookups
 - [DNS over TLS \(DoT\)](#) - A security protocol for encrypting and wrapping Domain Name System (DNS) queries and answers via the Transport Layer Security (TLS) protocol
 - [DNS over HTTPS \(DoH\)](#) - A protocol for performing remote Domain Name System (DNS) resolution via the HTTPS protocol
- Tools & Libraries
 - [OpenDKIM](#) - A community effort develops and maintains a C library for producing DomainKeys Identified Mail-aware applications and an open-source filter for providing the service

Governance, Risk, and Compliance (GRC)

Relevant DSS-P Skills

- 1. Business Transformation > 1.3 Management of Transformation Activities > Risk & Compliance
- 4. Security > 4.1 Security Management > Security Management
- 4. Security > 4.1 Security Management > Privacy Protection

Data Governance

- [Unity Catalog](#) - A universal catalog for data and AI that provides interoperability, openness, and unified governance across various formats and platforms
- [Microsoft Purview](#) - A unified approach to help organizations secure and govern data across their heterogeneous data estate
- [Amazon DataZone](#) - A data management service that makes it faster and easier for customers to catalog, discover, share, and govern data stored across AWS, on premises, and third-party sources

AI Governance & Security

- [ISO/IEC 42001](#) - The world's first AI management system standard, specifying requirements for establishing, implementing, maintaining and continually improving an Artificial Intelligence Management System (AIMS) within organizations
- [METR](#) - A research nonprofit that scientifically measures whether and when AI systems might threaten catastrophic harm to society
- [Microsoft Agent 365](#) - A centralized governance and management platform designed to oversee autonomous AI agents within an enterprise environment

Regulations & Standards

- Laws & Regulations
 - [General Data Protection Regulation \(GDPR\)](#) - The toughest privacy and security law in the world
 - Data subject requests (DSR)
 - Breach notification
 - Data protection impact assessment (DPIA)
 - [California Consumer Privacy Act \(CCPA\)](#) - A state statute intended to enhance privacy rights and consumer protection for residents of California
 - [Cyber Resilience Act](#) - A regulation that aims to safeguard consumers and businesses buying or using products or software with a digital component by introducing mandatory cybersecurity requirements
- Security & Privacy Frameworks
 - [NIST SP 800-53](#) - A catalog of security and privacy controls for all U.S. federal information systems except those related to national security
 - [OSCAL](#) - The Open Security Controls Assessment Language, a NIST-led initiative that provides open, machine-readable formats (XML, JSON, YAML) to automate security and compliance processes
 - [ISO/IEC 27001](#) - The international standard for information security management systems
- Industry & Audit Standards
 - [PCI-DSS](#) - The global standard for payment card data security
 - [SOC 2](#) - A voluntary compliance standard for service organizations which specifies how organizations should manage customer data

- [FIPS 140-2](#) - A U.S. government computer security standard used to approve cryptographic modules
- Hardening & Implementation Guides
 - [Security Technical Implementation Guides \(STIGs\)](#) - The configuration standards for DOD IA and IA-enabled devices/systems
 - [CIS Controls and Benchmarks](#) - A publication of best practice guidelines for computer security
 - [NIST SP 800-190](#) - The Application Container Security Guide, which explains the potential security concerns associated with the use of containers and provides recommendations for addressing these concerns

Vulnerability Management & Reporting

- [Bug bounty program](#) - A monetary reward given to ethical hackers for successfully discovering and reporting a vulnerability or bug to the application's developer
- Identifiers & Enumerations
 - [CVE \(Common Vulnerabilities and Exposures\)](#) - A system that provides a reference-method for publicly known information-security vulnerabilities and exposures
 - [CWE \(Common Weakness Enumeration\)](#) - A category system for software weaknesses and vulnerabilities
 - [OSV \(Open Source Vulnerability\)](#) - A vulnerability database and triage infrastructure for open source projects
- Scoring & Prioritization
 - [CVSS \(Common Vulnerability Scoring System\)](#) - A free and open industry standard for assessing the severity of computer system security vulnerabilities
 - [EPSS \(Exploit Prediction Scoring System\)](#) - A data-driven framework for estimating the probability that a software vulnerability will be exploited in the wild
 - [KEV \(Known Exploited Vulnerabilities\)](#) - A catalog that contains vulnerabilities that have been exploited in the wild
 - [SSVC \(Stakeholder-Specific Vulnerability Categorization\)](#) - A vulnerability management methodology that assesses the risk a vulnerability poses to an organization and provides a corresponding decision-making process
- Protocols & Databases
 - [Security Content Automation Protocol \(SCAP\)](#) - A multi-purpose framework of specifications that supports automated configuration, vulnerability and patch

checking

- [NVD \(U.S. National Vulnerability Database\)](#) - The U.S. government repository of standards based vulnerability management data
- [SARIF](#) - A standard format for the output of static analysis tools

System & Personal Security

Relevant DSS-P Skills

- 4. Security > 4.1 Security Management > Security System Construction and Operation
- 4. Security > 4.2 Security Technology > Secure Design, Development, and Construction

OS & Endpoint Security

- [Address space layout randomization \(ASLR\)](#) - A computer security technique involved in memory protection that prevents exploitation of memory vulnerabilities by randomly arranging the address space positions of key data areas of a process
- [W^X](#) - A security feature that ensures that every page in a process's address space is either writable or executable, but not both
- [Control-flow integrity](#) - A general term for computer security techniques that prevent a wide variety of malware attacks from redirecting the flow of execution of a program
- [TPM \(Trusted Platform Module\)](#) - A specification for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys
- Linux Mandatory Access Control
 - [SELinux](#) - A set of kernel modifications and user-space tools that have been added to various Linux distributions
 - [AppArmor](#) - A Linux kernel security module that allows the system administrator to restrict programs' capabilities with per-program profiles
 - [bubblewrap](#) - A low-level unprivileged sandboxing tool used by Flatpak and similar projects
- Linux Fine-grained Access Control
 - [Linux capabilities](#) - A feature that grants some of the power of the superuser to a process, while not granting all of them

- General Scanning
 - [OpenSCAP](#) - An open source implementation of the Security Content Automation Protocol (SCAP)
 - [Lynis](#) - A security auditing tool for systems running Linux, macOS, or Unix-based operating system

Personal Security Tools

- Password Managers
 - [1Password](#) - A password manager and secure vault platform for identities, credentials, and secrets across humans and AI agents
 - [pass](#) - The standard unix password manager
 - [gokey](#) - A simple vaultless password manager in Go
 - [Buttercup](#) - A free, open-source and cross-platform password manager

05 - Data Science & Engineering

Foundational Concepts

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Understanding and Utilization of Data and AI

General Data Concepts & Principles

- [Data](#) - Any sequence of one or more symbols; datum is a single symbol of data
- [Metadata](#) - The data that provides information about other data, but not the content of the data
- [Big data](#) - The data sets that are too large or complex to be dealt with by traditional data-processing application software
- [Unstructured data](#) - The information that either does not have a pre-defined data model or is not organized in a pre-defined manner
- [Data model](#) - An abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities
 - [Entity-relationship model](#) - An abstract description of interrelated things of interest in a specific domain of knowledge
- [Data orientation](#) - A perspective of data that emphasizes the data itself, rather than

the applications that use the data

- **DIKW pyramid** - A class of models representing purported structural and/or functional relationships between data, information, knowledge, and wisdom
- **Garbage in, garbage out** - A concept in computer science and information and communications technology that the quality of the output is determined by the quality of the input
- **Data cleansing** - The process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database
- **Data lifecycle management** - A policy-based approach to managing the flow of an information system's data throughout its life cycle
- **Master data** - The data about the business entities that provide context for business transactions
- **Master data management** - A technology-enabled discipline in which business and IT work together to ensure the uniformity, accuracy, stewardship, semantic consistency and accountability of the enterprise's official shared master data assets
- **Data quality** - A measure of the condition of data based on factors such as accuracy, completeness, consistency, reliability and whether it's up to date
- **Single source of truth** - The practice of structuring information models and associated data schema such that every data element is mastered (or edited) in only one place

Core Data Engineering & Database Concepts

- **Concurrency control** - The mechanism ensuring that correct results for concurrent operations are generated efficiently
- **CRUD operations** - The four basic operations of persistent storage: create, read, update, and delete
- **Shard** - A horizontal partition of data in a database or search engine
- **ETL** - A three-phase process where data is extracted from an input source, transformed, and loaded into an output data container
- **ELT** - A data integration process where raw data is moved from a source system to a destination resource, such as a data warehouse, and then transformed for use
- **Data pipeline** - A set of data processing elements connected in series, where the output of one element is the input of the next one
- **Data governance** - A data management concept concerning the capability that enables an organization to ensure that high data quality exists throughout the complete lifecycle of the data

- [Data lineage](#) - The process of understanding, recording, and visualizing data as it flows from data sources to consumption
- [Online transaction processing \(OLTP\)](#) - A type of data processing that consists of executing a number of transactions occurring concurrently
- [Online analytical processing \(OLAP\)](#) - An approach to answering multi-dimensional analytical queries swiftly in computing
- [Search engine indexing](#) - The collecting, parsing, and storing of data to facilitate fast and accurate information retrieval

Data Governance, Quality & Architecture

- [Data Catalog](#) - A centralized metadata repository that helps organizations manage and discover data assets
- [Data Stewardship](#) - A set of practices and processes for managing an organization's data assets to ensure quality, security, and compliance
- [Data Privacy](#) - The right and ability of an individual to determine what happens to information about themselves
- [Data Security](#) - The process of securing digital information to protect it from online threats
- [ISO 8000](#) - The international standard for data quality and master data
- [Data Contract](#) - An explicit agreement on data structure, quality, and semantics between data producers and consumers
- [Schema Evolution](#) - The process of modifying a database schema while maintaining compatibility with existing data and applications
- [Dimensional Modeling](#) - A database design technique used to optimize data warehouses for analytical queries using facts and dimensions

Data Science Toolkit

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Mathematical Statistics, Multivariate Analysis, and Data Visualization

Programming Languages & Libraries

- [Python](#) - A programming language that lets you work quickly and integrate systems more effectively

- [Awesome Python](#) - A curated list of awesome Python frameworks, libraries, tools, and resources
- [Pandas](#) - A fast, powerful, flexible and easy to use open source data analysis and manipulation tool
- [Polars](#) - A blazingly fast DataFrame library for manipulating structured data
- [Narwhals](#) - A lazy-first, type-agnostic, and framework-agnostic dataframe library in Python
- [NumPy](#) - The fundamental package for scientific computing with Python
- [SciPy](#) - Fundamental algorithms for scientific computing in Python
- [SymPy](#) - A Python library for symbolic mathematics
- [SageMath](#) - A free open-source mathematics software system licensed under the GPL
- [statsmodels](#) - A Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration
- [R](#) - A free software environment for statistical computing and graphics
 - [Tidyverse](#) - An opinionated collection of R packages designed for data science
- [GNU Octave](#) - A high-level language, primarily intended for numerical computations
- [Wolfram Language](#) - A symbolic language, deliberately designed with the breadth and unity needed to develop powerful programs quickly

Specialized & Scientific Tools

- [latexify](#) - A Python package to compile a fragment of Python source code to a corresponding LaTeX expression
- [handcalcs](#) - A Python library to render Python calculation code automatically in Latex, but in a manner that mimics how one might format their calculation if it were written with a pencil
- [NetworkX](#) - A Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks
- [JAX](#) - A Python library for accelerator-oriented array computation and program transformation

Data Sources & Geospatial

- [GeoLite2](#) - A set of free geolocation and ASN data in downloadable database and web service formats

Spreadsheet & Collaborative Data Platforms

- [Microsoft Excel](#) - The industry-leading spreadsheet software program and a powerful data visualization and analysis tool
- [Grist](#) - A relational spreadsheet that combines the familiar interface of a spreadsheet with the power and structure of a relational database
- [NocoBase](#) - A scalability-first, open-source no-code platform designed for building complex business applications and internal tools
- [NocoDB](#) - An open-source, no-code platform that turns any database into a smart spreadsheet, providing a collaborative interface for relational databases
- [Airtable](#) - A platform that combines the flexibility of a spreadsheet with the power of a database to help teams manage their work

Interactive Computing Environments

- [JupyterLab](#) - A web-based interactive development environment for notebooks, code, and data
- [Jupyter Notebook](#) - The original web application for creating and sharing computational documents
 - [VSCode Jupyter Extension](#) - A VS Code extension that provides basic notebook support for language kernels supported in the environment
- [nbviewer](#) - A simple way to share Jupyter Notebooks
- [R Markdown](#) - An authoring framework that helps you create dynamic analysis documents combining code, rendered output, and prose
- [Wolfram Notebooks](#) - A powerful environment for exploration and communication, combining text, literate programming, graphics and custom interactive elements
- [Voila](#) - A tool that turns Jupyter notebooks into standalone web applications

Data Visualization

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Mathematical Statistics, Multivariate Analysis, and Data Visualization

Common Chart Types

- [Histogram](#) - A representation of the distribution of numerical data

- [Scatter plot](#) - A type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data
- [Box plot](#) - A method for graphically demonstrating the locality, spread and skewness groups of numerical data through their quartiles
- [Error bar](#) - A graphical representation of the variability of data used on graphs to indicate the uncertainty in a reported measurement
- [Heat map](#) - A technique that shows magnitude of a phenomenon as color in two dimensions
- [Choropleth map](#) - A type of thematic map in which a set of pre-defined areas is colored or patterned in proportion to a statistical variable
- [Proportional symbol map](#) - A type of thematic map that uses symbols that vary in size to represent a quantitative variable
- [Tag cloud](#) - A novelty visual representation of text data

Visualization Tools & Libraries

- Python Libraries
 - [matplotlib](#) - A comprehensive library for creating static, animated, and interactive visualizations in Python
 - [seaborn](#) - A Python data visualization library based on matplotlib
 - [Plotly](#) - The interactive, open-source, and browser-based graphing library for Python (includes Plotly Express)
 - [WordCloud for Python](#) - A little word cloud generator in Python
- JavaScript Libraries
 - [D3](#) - The JavaScript library for bespoke data visualization
 - [GoJS](#) - A JavaScript library that lets you easily create interactive diagrams in web browsers
 - [Chart.js](#) - A simple yet flexible JavaScript charting library for the modern web
 - [Recharts](#) - A composable charting library built on React components
 - [Tabulator](#) - An easy to use, simple to code, fully featured, interactive JavaScript library for creating tables and data grids
- Grammars & Other
 - [gnuplot](#) - A portable command-line driven graphing utility
 - [ggplot2](#) - A system for declaratively creating graphics, based on The Grammar of Graphics

- [Vega](#) - A visualization grammar, a declarative language for creating, saving, and sharing interactive visualization designs
- [Vega-Lite](#) - A high-level grammar of interactive graphics

Dashboarding & Web Apps

- [Dash](#) - The original low-code framework for rapidly building data apps in Python, R, Julia, and F#
- [Panel](#) - A powerful Python library that lets you create interactive web apps and dashboards
- [Streamlit](#) - A faster way to build and share data apps

Distributed Systems

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.3 Data Management > Data Engineering (Design, Collection, Integration, Provision)

Distributed Computing Principles

- [Distributed computing](#) - A field of computer science that studies such systems
- [Single point of failure](#) - A part of a system that, if it fails, will stop the entire system from working
- [Fault tolerance](#) - The property that enables a system to continue operating properly in the event of the failure of some of its components
- [Load balancing](#) - The process of distributing a set of tasks over a set of resources, with the aim of making their overall processing more efficient
- [Fallacies of distributed computing](#) - A set of assertions describing false assumptions that programmers new to distributed applications invariably make
- [Byzantine fault](#) - A condition of a distributed system, where components may fail and there is imperfect information about whether a component has failed
 - [Consensus](#) - A fault-tolerant mechanism that is used in distributed systems to achieve the necessary agreement on a single data value among distributed processes or systems
- [CAP theorem](#) - A theorem stating that any distributed data store can provide only two of the following three guarantees: Consistency, Availability, and Partition tolerance

- [BASE properties](#) - A database model that prioritizes availability over consistency
- [Amdahl's law](#) - A formula limiting the speedup of a task as resources are added to the system executing that task

Consensus & Replication Strategies

- [Raft Consensus Algorithm](#) - A consensus algorithm designed to be more understandable than Paxos, enabling safe state machine replication across clusters
- [Paxos Algorithm](#) - A family of protocols for solving consensus in a network of unreliable or asynchronous processors
- [Data Replication](#) - The frequent electronic copying of data from a computer or server to another location, computer, or server
 - Master-Slave Replication - A pattern where one primary node accepts writes and slaves replicate data
 - [Consensus](#) - A fault-tolerant mechanism that is used in distributed systems to achieve the necessary agreement on a single data value among distributed processes or systems

Distributed Patterns & Observability

- [Circuit Breaker Pattern](#) - A design pattern to prevent cascading failures in distributed systems
- [Distributed Tracing](#) - A method for profiling and monitoring applications, especially those built using microservices architecture
- [Event Sourcing](#) - A pattern where all changes to application state are stored as a sequence of immutable events

Distributed Storage Systems

- Distributed File Systems
 - [HDFS](#) - A distributed file system designed to run on commodity hardware
 - [IPFS](#) - A peer-to-peer hypermedia protocol designed to make the web faster, safer, and more open
 - [Kubo](#) - A Go implementation of IPFS
- [Object storage](#) - A computer data storage architecture that manages data as objects
 - [Amazon S3](#) - An object storage service offering industry-leading scalability, data availability, security, and performance
 - [Azure Blob Storage](#) - The Microsoft's object storage solution for the cloud,

optimized for storing massive amounts of unstructured data

- [Azure Data Lake Storage \(ADLS\)](#) - A scalable and secure data lake for high-performance analytics workloads
- [Google Cloud Storage](#) - A RESTful online file storage web service for storing and accessing data on Google Cloud Platform infrastructure
- [Cloud Storage for Firebase](#) - The service letting you upload and share user generated content, such as images and video
- [Supabase Storage](#) - The service making it simple to store and serve large files like photos and videos
- Self-hosted (advanced)
 - [Ceph](#) - An open-source, distributed storage system
 - [MinIO](#) - A high-performance, S3 compatible object store
- Tooling
 - [s5cmd](#) - A very fast S3 and local filesystem execution tool
 - [Rclone](#) - A command-line program to manage files on cloud storage
 - [Azure Storage Explorer](#) - A standalone app making it easy to work with Azure Storage data on Windows, macOS, and Linux
 - [Azurite](#) - An open-source Azure Storage emulator

Mathematics & Statistics

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Mathematical Statistics, Multivariate Analysis, and Data Visualization

Base Mathematics

- [Algebra](#) - A branch of mathematics that deals with abstract systems, known as algebraic structures, and the manipulation of expressions within those systems
 - [Boolean algebra](#) - A branch of algebra that differs from elementary algebra in that the values of the variables are the truth values true and false, usually denoted by 1 and 0, and it uses logical operators such as conjunction (and), disjunction (or), and negation (not)
 - [Elementary algebra](#) - A branch of mathematics that encompasses the basic concepts of algebra
 - [Equation](#) - A mathematical formula that expresses the equality of two

expressions, by connecting them with the equals sign =

- **Logarithm** - The exponent by which another fixed value, the base, must be raised to produce that number
- **Abstract algebra** - The study of algebraic structures, which are sets with specific operations acting on their elements
- **Linear algebra** - The branch of mathematics concerning linear equations, linear maps, and their representations in vector spaces and through matrices
 - **Vector space** - A set whose elements, often called vectors, can be added together and multiplied ("scaled") by numbers called scalars
 - **Matrix** - A rectangular array of numbers or other mathematical objects with elements or entries arranged in rows and columns, usually satisfying certain properties of addition and multiplication
 - **Sparse matrix** - A matrix in which most of the elements are zero
 - **Rank** - The dimension of the vector space generated (or spanned) by its columns
 - **Determinant** - A scalar-valued function of the entries of a square matrix
- **Calculus** - The mathematical study of continuous change, in the same way that geometry is the study of shape and algebra is the study of generalizations of arithmetic operations
 - **Differential calculus** - A subfield of calculus that studies the rates at which quantities change
 - **Integral calculus** - The continuous analog of a sum, and is used to calculate areas, volumes, and their generalizations
 - **Differential equation** - An equation that relates one or more unknown functions and their derivatives
- **Geometry** - A branch of mathematics concerned with properties of space such as the distance, shape, size, and relative position of figures
 - **Trigonometry** - A branch of mathematics concerned with relationships between angles and side lengths of triangles
 - **Coordinate system** - A system that uses one or more numbers, or coordinates, to uniquely determine and standardize the position of the points or other geometric elements on a manifold such as Euclidean space
 - **Euclidean distance** - The length of the line segment between two points in a Euclidean space
- **Category theory** - A general theory of mathematical structures and their relations

- [Functor](#) - A mapping between categories
- [Root mean square](#) - The square root of the mean of the squares of a set of numbers
- Transforms
 - [Discrete cosine transform](#) - A transform that expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies
 - [Discrete Fourier transform](#) - A discrete version of the Fourier transform that converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT)
- Related Resources
 - [NIST Digital Library of Mathematical Functions](#) - The definitive reference for the special functions of applied mathematics
 - [Notations](#) - A list of notations used in the library

Probability & Information Theory

- [Probability theory](#) - The branch of mathematics concerned with probability
 - [Bayes' theorem](#) - A mathematical rule for inverting conditional probabilities, allowing the probability of a cause to be found given its effect
 - [Central limit theorem \(CLT\)](#) - A theorem that states that, under appropriate conditions, the distribution of a normalized version of the sample mean converges to a standard normal distribution
- [Information theory](#) - A scientific study of the quantification, storage, and communication of digital information
 - [Entropy](#) - The average level of 'information', 'surprise', or 'uncertainty' inherent in a random variable's possible outcomes

Statistics & Numerical Methods

- [Statistics](#) - A discipline that concerns the collection, organization, analysis, interpretation, and presentation of data
 - [Sampling](#) - The selection of a subset of individuals from within a statistical population to estimate characteristics of the whole population
 - [Sampling error](#) - The error caused by observing a sample instead of the whole population
 - [Errors and residuals](#) - The measures of the deviation of an observed value of an element of a statistical sample from its "true value"

- [Frequency](#) - The number of times the observation has occurred or been recorded in an experiment or study
 - [Contingency table](#) - A type of table in a matrix format that displays the multivariate frequency distribution of the variables
- [Confounding](#) - A variable that influences both the dependent variable and independent variable, causing a spurious association
- [Standard deviation](#) - A measure of the amount of variation of the values of a variable about its average
- [Root mean square deviation](#) - The square root of the average of the squared differences between the predicted values and the actual values
- [F-score](#) - A measure of predictive performance in statistical analysis of binary classification and information retrieval systems
- [Correlation](#) - A kind of statistical relationship between two random variables or bivariate data
 - [Pearson correlation coefficient](#) - A correlation coefficient that measures linear correlation between two sets of data
- [Hypothesis testing](#) - A method of statistical inference used to decide whether the data provide sufficient evidence to reject a particular hypothesis
 - [Null hypothesis](#) - A typical statistical theory which suggests that no statistical relationship and significance exists in a set of given single observed variable, between two sets of observed data and measured phenomena
 - [Confidence interval \(CI\)](#) - A range of values which is likely to contain (in repeated sampling) the true value of an unknown statistical parameter, such as a population mean
 - [P-value](#) - The probability of obtaining test results at least as extreme as the result actually observed, under the assumption that the null hypothesis is correct
- Numerical methods
 - [Significant figures](#) - The specific digits within a number that is written in positional notation that carry both reliability and necessity in conveying a particular quantity
- Resources
 - [Openstax Introductory Statistics](#) - An open-source textbook for introductory statistics courses
 - [OpenIntro Statistics](#) - A dynamic take on the traditional curriculum, being successfully used at Community Colleges to the Ivy League

Data Formats & Architecture

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.3 Data Management > Data Engineering (Design, Collection, Integration, Provision)
- 2. Data Preparation & Utilization > 2.3 Data Management > Improvement of Data Quality and Safety

Data Formats & Table Formats

- [Apache Parquet](#) - An open source, column-oriented data file format designed for efficient data storage and retrieval
- [Apache ORC](#) - The smallest, fastest columnar storage for Hadoop workloads
- [Apache Arrow](#) - A universal columnar format and multi-language toolbox for fast data interchange and in-memory analytics
- [BSON](#) - A binary-encoded serialization of JSON-like documents
- [Apache Avro](#) - The leading serialization format for record data, and first choice for streaming data pipelines
- [Delta Lake](#) - An open-source storage framework that enables building a format agnostic Lakehouse architecture with compute engines
- [Apache Iceberg](#) - The open table format for huge analytic datasets
- [Apache Hudi](#) - The Streaming Data Lake Platform

Data Architectures & Methodologies

- [Data warehouse](#) - A system used for reporting and data analysis and is a core component of business intelligence
- [Data lake](#) - A system or repository of data stored in its natural/raw format, usually object blobs or files
- [Data lakehouse](#) - A new, open architecture that combines the best elements of data lakes and data warehouses
- [Medallion Architecture](#) - A data design pattern used to logically organize data in a lakehouse
- [CRISP-DM](#) - An open standard process model that describes common approaches used by data mining experts
- [PPDAC \(Problem, Plan, Data, Analysis, Conclusion\)](#) - A well-established approach to

statistical literacy, a cyclical framework for solving real-world challenges using data

- [Data architecture](#) - A set of models, policies, rules, and standards that govern which data is collected and how it is stored, arranged, integrated, and put to use in data systems and in organizations
- [DAMA-DMBOK](#) - The DAMA Guide to the Data Management Body of Knowledge, outlining frameworks and terminology across thirteen functional areas of managing data

Data Governance & Metadata Management

- [Apache Atlas](#) - A scalable and extensible set of core foundational governance services that enable enterprises to meet compliance requirements
- [Collibra](#) - An enterprise data governance platform providing a common language for data management
- [Informatica Metadata Manager](#) - A comprehensive metadata management solution for enterprise data governance
- [OpenMetadata](#) - An open-source metadata management platform for data discovery, governance, and collaboration

Data Quality & Validation

- [Great Expectations](#) - A Python library for defining, documenting, and testing data quality
- [Apache Griffin](#) - A data quality solution built on Apache Spark and Apache Hadoop for distributed data quality measurement
- [Soda](#) - A data quality monitoring solution that integrates with modern data stacks

Data Versioning & Schema Management

- [Schema Registry](#) - A hosted schema management service that centralizes schemas for Kafka topics
- Git-based Schema Management - Using Git repositories to version control database schemas
- [DBT Contracts](#) - Explicit data contracts defining input and output data requirements

Relational Databases (SQL)

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.3 Data Management > Data Engineering (Design, Collection, Integration, Provision)

SQL Fundamentals

- Foundational Concepts
 - [Relational model](#) - An approach to managing data using a structure and language consistent with first-order predicate logic
 - [ACID properties](#) - A set of properties of database transactions intended to guarantee data validity despite errors, power failures, and other mishaps
 - Atomicity, Consistency, Isolation, and Durability
 - [Codd's Twelve Rules](#) - A set of thirteen rules proposed by Edgar F. Codd to define what is required from a database management system in order for it to be considered relational
 - [Database normalization](#) - The process of organizing columns (attributes) and tables (relations) of a relational database to minimize data redundancy
- Languages & Dialects
 - [Structured Query Language \(SQL\)](#) - A domain-specific language used for managing data held in a relational database management system
 - Command Categories
 - DDL - Data Definition Language
 - DQL - Data Query Language
 - DML - Data Manipulation Language
 - DCL - Data Control Language
 - TCL - Transaction Control Language
 - [SQL Join](#) - A clause that combines columns from one or more tables in a relational database
 - [Aggregate function](#) - A function where the values of multiple rows are grouped together to form a single summary value
 - [Transact-SQL](#) - The proprietary extension to SQL used to program and manage SQL Server

Database Management Systems (DBMS)

- Client-Server RDBMS
 - [PostgreSQL](#) - An object-relational database management system (ORDBMS)

based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department

- [MySQL](#) - The most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation
- [MariaDB community Server](#) - The open source relational database that is a community-developed fork of MySQL
- Distributed SQL
 - [TiDB](#) - An open-source distributed SQL database that supports Hybrid Transactional and Analytical Processing (HTAP) workloads
- Embedded / In-Process
 - [SQLite](#) - A C-language library that implements a small, fast, self-contained, high-reliability, and full-featured database engine
 - [PGLite](#) - A WASM build packaged into a TypeScript/JavaScript client library, that enables you to run the database in the browser, Node.js and Bun
 - [DuckDB](#) - An in-process SQL OLAP database management system
- Storage Engines
 - [Storage Engine](#) - A software component that a database management system uses to create, read, update and delete (CRUD) data from a database
 - [InnoDB](#) - A transactional storage engine for MySQL and MariaDB

Cloud & Managed Services

- Managed Database Services
 - [Amazon RDS](#) - A collection of managed services that makes it simple to set up, operate, and scale databases in the cloud
 - [Amazon Aurora](#) - A fully managed relational database engine offering high performance and availability at global scale for PostgreSQL, MySQL, and DSQL
 - [Azure SQL Database](#) - An intelligent, scalable, relational database service built for the cloud
 - [Azure HorizonDB](#) - A fully-managed, cloud-native PostgreSQL-compatible database service engineered for high-throughput, AI-powered applications with scale-out architecture supporting up to 3,072 vCores and 128 TB storage
 - [Google Cloud SQL](#) - A fully-managed database service that helps you set up, maintain, manage, and administer your relational databases on Google Cloud
 - [Neon](#) - A serverless, fault-tolerant, and scalable Postgres with a generous free tier

- [Turso](#) - A SQLite-compatible database built on a ground-up rewrite of SQLite, lightweight enough to multiply and fast enough to run anywhere

Connectivity & Tooling

- Connectivity APIs & ORMs
 - [Connection pool](#) - A cache of database connections maintained so that the connections can be reused when future requests to the database are required
 - [ODBC](#) - A standard application programming interface for accessing database management systems
 - [JDBC](#) - An API that allows access to virtually any tabular data source from the Java programming language
 - [Jdbi](#) - A library that provides a more idiomatic way to use the Java Database Connectivity API
 - [Object-Relational Mapping \(ORM\)](#) - A programming technique for converting data between incompatible type systems using object-oriented programming languages
 - [Prisma](#) - A next-generation ORM that makes it easy to build reliable and scalable applications with databases
 - [Drizzle ORM](#) - A lightweight and performant TypeScript ORM with developer experience in mind
 - [Hibernate](#) - An object-relational mapping tool for the Java programming language
 - [SQLAlchemy](#) - The Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL
 - [GORM](#) - The fantastic ORM library for Golang aims to be developer friendly
 - [XORM](#) - A Simple and Powerful ORM for Go
 - [Diesel](#) - A Safe, Extensible ORM and Query Builder for Rust
- Developer Libraries & Drivers
 - [Vanna.AI](#) - A Python package that uses retrieval augmentation to help generate accurate SQL queries using LLMs
 - [Psycopg](#) - The most popular PostgreSQL adapter for the Python programming language
- Database Clients & IDEs
 - [pgAdmin](#) - The most popular and feature rich Open Source administration and development platform for PostgreSQL

- [SSMS \(SQL Server Management Studio\)](#) - An integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database
- [DB Browser for SQLite](#) - A high quality, visual, open source tool to create, design, and edit database files compatible with SQLite
- [Azure Data Studio](#) - A modern open-source, cross-platform hybrid data analytics tool designed to simplify the data landscape
- [Beekeeper Studio](#) - A modern, easy to use, and good looking SQL editor and database manager
- Command-Line & Deployment Utilities
 - [sqlcmd utility](#) - A command-line utility for ad hoc, interactive execution of Transact-SQL statements and scripts and for automating T-SQL scripting tasks
 - [sqlpackage](#) - A command-line utility that automates several database development tasks
 - [DAC \(Data-tier Applications\)](#) - A logical database management concept that defines all of the SQL Server objects associated with a user's database
 - [pgroll](#) - A zero-downtime, reversible, schema migration tool for PostgreSQL
- Monitoring & Analysis
 - [pgBadger](#) - A PostgreSQL log analyzer built for speed with fully detailed reports and professional rendering
- Performance Benchmarks
 - [TPC-H](#) - A decision support benchmark that evaluates database systems through business-oriented ad-hoc queries and data modifications against large data volumes
 - [TPC-DS](#) - A decision support benchmark that simulates complex database queries and big data environments to evaluate system performance and price/performance metrics

NoSQL & Specialized Databases

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.3 Data Management > Data Engineering (Design, Collection, Integration, Provision)

NoSQL Data Models

- [Object-relational impedance mismatch](#) - A set of conceptual and technical

difficulties that are often encountered when a relational database management system (RDBMS) is being used by a program written in an object-oriented programming language or style

- Document Databases
 - [MongoDB](#) - A document database designed for ease of application development and scaling
 - [DocumentDB](#) - A powerful, scalable open-source document database built for modern applications
- Key-value Stores
 - [etcd](#) - A distributed, reliable key-value store for the most critical data of a distributed system
 - [Redis](#) - An in-memory data store used by millions of developers as a cache, vector database, document database, streaming engine
 - [Dragonfly](#) - A drop-in Redis replacement
- Graph Databases
 - [Neo4j](#) - A high-speed graph database with unbounded scale, security, and data integrity
 - [Cypher](#) - A declarative query language for property graph databases
 - [LadybugDB](#) - An embedded columnar graph database built for highly regulated industries
- Wide-columns Databases
 - [Apache Cassandra](#) - An open source NoSQL distributed database
 - [Apache HBase](#) - The Hadoop database, a distributed, scalable, big data store
 - [ClickHouse](#) - A fast, open-source OLAP (Online Analytical Processing) database management system designed for real-time analytics

Vector & AI Databases

- Concepts
 - [HNSW \(Hierarchical Navigable Small Worlds\)](#) - A top-performing index for vector similarity search
- Vector Databases
 - [Pinecone](#) - A purpose-built vector database delivering relevant results at any scale
 - [pgvector](#) - An open-source vector similarity search for Postgres

- [ElasticSearch vector database](#) - The world's most widely deployed, open source vector database
- [Weaviate](#) - An open-source vector database that simplifies the development of AI applications
- [Milvus](#) - A high-performance open-source vector database built to handle billions of vectors
- [Chroma](#) - The AI-native open-source embedding database
- [Qdrant](#) - A high-performance vector search engine built entirely in Rust that helps developers build AI retrieval at any scale

Cloud NoSQL Services

- Multi-model Databases
 - [Azure Cosmos DB](#) - A fully managed, serverless distributed database for modern app development
 - [Amazon DynamoDB](#) - A fully managed, serverless, key-value NoSQL database designed to run high-performance applications at any scale
- Document Databases
 - [Cloud Firestore](#) - A cloud-hosted, NoSQL database that your Apple, Android, and web apps can access directly via native SDKs
- Graph Databases
 - [Amazon Neptune](#) - A fast, reliable, and fully managed graph database service that makes it easy to build and run applications that work with highly connected datasets
- Wide-columns Databases
 - [Google Cloud Bigtable](#) - A NoSQL wide-column database service for large analytical and operational workloads

Data Processing & Messaging

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.3 Data Management > Data Engineering (Design, Collection, Integration, Provision)

Enterprise Integration

- [Enterprise Integration Patterns](#) - A pattern language of 65 integration patterns that

helps developers design and build distributed applications or integrate existing ones

- [Apache Camel](#) - An open-source integration framework that empowers you to quickly and easily integrate various systems consuming or producing data

Message Queuing & Event Streaming

- Concepts
 - [Message Brokers](#) - An intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver
 - [Dead-letter queue](#) - A specialized queue used in message queuing systems to store messages that could not be delivered or processed successfully
- Messaging & Streaming Platforms (Software)
 - [Apache Kafka](#) - An open-source distributed event streaming platform
 - Apache Kafka Ecosystem
 - [Kafbat UI](#) - A versatile, fast, lightweight, and flexible web interface designed to monitor and manage Apache Kafka clusters
 - [RabbitMQ](#) - A reliable and mature messaging and streaming broker
- Cloud Services
 - [Amazon Kinesis](#) - A service making it easy to collect, process, and analyze real-time, streaming data
 - [Azure Event Hubs](#) - A highly scalable and reliable event streaming platform capable of ingesting millions of events per second
 - [Azure Service Bus](#) - A fully managed enterprise message broker with message queues and publish-subscribe topics

Batch Processing (ETL/ELT)

- Base Frameworks
 - [Apache Hadoop](#) - A framework that allows for the distributed processing of large data sets
 - [mrjob](#) - The easiest route to writing Python programs that run on the framework
 - [Apache Spark](#) - The unified engine for large-scale data analytics
 - [PySpark](#) - The Python API for the engine, allowing big data processing with

the language

- [RAY](#) - An open-source unified compute framework that makes it easy to scale AI and Python workloads
- [Joblib](#) - A set of tools to provide lightweight pipelining in Python
- Workflow Orchestration & ETL Tools (Software)
 - [Apache NiFi](#) - An easy to use, powerful, and reliable system to process and distribute data
 - [Apache Airflow](#) - A platform to programmatically author, schedule, and monitor workflows
 - [dbt](#) - A unified platform for delivering trusted data that empowers teams to deliver reliable, governed data at scale
 - [Dagu](#) - A local-first workflow engine that provides a declarative, file-based, and self-contained platform to orchestrate tasks from a single binary that scales from a laptop to a distributed cluster
- Managed ETL & Data Integration Services
 - [Azure Data Factory](#) - The cloud ETL service for scale-out serverless data integration and data transformation
 - [AWS Glue](#) - A serverless data integration service that makes it easy to discover, prepare, move, and integrate data from multiple sources
 - [Google Cloud Data Fusion](#) - A fully managed, cloud-native data integration service that helps users efficiently build and manage ETL/ELT data pipelines

Stream Processing

- Stream Processing Engines (Software)
 - [Spark Structured Streaming](#) - A scalable and fault-tolerant stream processing engine built on the Spark SQL engine
 - [Apache Storm](#) - A free and open source distributed realtime computation system
 - [Apache Flink](#) - A framework and distributed processing engine for stateful computations over unbounded and bounded data streams
- Cloud Services
 - [Google Cloud Dataflow](#) - A fully managed streaming analytics service that minimizes latency, processing time, and cost through autoscaling and batch processing

Data Analytics & Search

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Understanding and Utilization of Data and AI

Search Engines & Platforms

- Web Search Engines
 - [Google Search](#) - The search engine that allows you to search the world's information, including webpages, images, videos and more
 - [DuckDuckGo](#) - The search engine that doesn't track you
- Answer Engines
 - [Wolfram|Alpha](#) - A computational knowledge engine that computes expert-level answers using breakthrough algorithms, knowledgebase and AI technology
 - [Perplexity AI](#) - An AI-powered answer engine that provides accurate, trusted, and real-time answers to any question
- Search Platforms and Tools
 - [Azure AI Search](#) - A fully managed, cloud-hosted service that unifies access to enterprise and web content for AI-powered search and retrieval-augmented generation
 - [Reciprocal Rank Fusion \(RRF\)](#) - An algorithm that evaluates the search scores from multiple, previously executed queries to produce a unified result set
 - [BM25 relevance scoring](#) - The Okapi BM25 ranking function used to compute the relevance scores of matching documents in full text search
 - [ElasticSearch](#) - An open source distributed, RESTful search and analytics engine, scalable data store, and vector database
 - [Painless](#) - A simple, secure scripting language designed specifically for use with the engine
 - [ES|QL](#) - A piped language that allows you to filter, transform, and analyze data stored in the engine
 - [Kibana](#) - The open source interface to query, analyze, visualize, and manage your data stored in the engine
 - [Kibana Query Language](#) - A simple text-based query language for filtering data
 - [Apache Solr](#) - The popular, blazing-fast, open source enterprise search platform

built on Apache Lucene

- [Apache Lucene](#) - A Java library providing powerful indexing and search features
- [Faiss](#) - A library for efficient similarity search and clustering of dense vectors
- [Meilisearch](#) - A lightning-fast search engine that fits effortlessly into your apps, websites, and workflow
- [TypeSense](#) - A lightning-fast, open source, search-as-you-type engine for building delightful search experiences

Analytics Engines & Platforms

- Software & Managed Services
 - [Apache Hive](#) - A distributed, fault-tolerant data warehouse system that enables analytics at a massive scale
 - [Presto](#) - A distributed SQL query engine designed for fast, reliable, and efficient analytics at any scale
 - [Trino](#) - A distributed SQL query engine designed to query large data sets distributed over one or more heterogeneous data sources
 - [Amazon EMR](#) - A cloud big data platform for running large-scale distributed data processing jobs, interactive SQL queries, and machine learning applications
 - [Amazon Redshift](#) - A fully managed, petabyte-scale data warehouse service in the cloud
 - [Amazon Athena](#) - An interactive query service that makes it easy to analyze data directly in Amazon S3 and other data stores using standard SQL
 - [Databricks](#) - The platform that allows your entire organization to use data and AI
 - [Declarative Automation Bundles](#) - A tool to facilitate the adoption of software engineering best practices, including source control, code review, testing, and continuous integration and delivery (CI/CD), for data and AI projects, formerly known as Databricks Asset Bundles
 - [Snowflake](#) - The AI Data Cloud that mobilizes data with near-unlimited scale to power analytics, applications, and AI in a single fully managed platform
 - [Microsoft Fabric](#) - An end-to-end analytics solution with full-service capabilities including data movement, data lakes, data engineering, data integration, data science, real-time analytics, and business intelligence
 - [Microsoft OneLake](#) - A single, unified, logical data lake for your whole organization

- [Real-Time Intelligence](#) - A service that extracts insights from streaming data in motion with end-to-end solutions for ingestion, transformation, storage, analytics, visualization, and real-time actions on time-based events
- [Rayfin CLI](#) - A command-line tool for creating, deploying, and managing Fabric applications with project scaffolding, remote deployment, and configuration management capabilities
- [Lakehouse vs Data Warehouse](#) - A guide for choosing between a lakehouse and a data warehouse based on data volume, structure, and processing requirements
- [Azure Synapse Analytics](#) - An enterprise analytics service that accelerates time to insight across data warehouses and big data systems
- [Google Cloud BigQuery](#) - A fully managed, AI-ready data analytics platform that helps you maximize value from your data and is designed to be multi-engine, multi-format, and multi-cloud
- [Amazon QuickSight](#) - An AI-powered business intelligence service that enables users to analyze data, create visualizations, and gain insights from various enterprise data sources

Semantic Layer

- [Cube](#) - The agentic analytics platform to deploy AI agents to model, analyze, and report on your data
- [Open Semantic Interchange \(OSI\)](#) - The universal standard for semantic model exchange enabling semantic metadata interchange across analytics, AI, and BI platforms

06 - AI, Machine Learning & LLM

Foundations of AI & Machine Learning

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Understanding and Utilization of Data and AI
- AI Paradigms
 - [Symbolic AI](#) - The collection of all methods in artificial intelligence research that are based on high-level symbolic (human-readable) representations of problems, logic and search

- **Neuro-symbolic AI** - A subfield of artificial intelligence that integrates neural methods with symbolic methods to combine the strengths of both approaches, resulting in AI systems that can be trained from raw data while preserving explainability and explicit reasoning
 - **AlphaGeometry** - A neuro-symbolic system made up of a neural language model and a symbolic deduction engine, which work together to find proofs for complex geometry theorems
- **Generative AI** - A subset of artificial intelligence that uses generative models to produce text, images, videos, or other forms of data
- **Causal AI** - A technique in artificial intelligence that builds a causal model and can thereby make inferences using causality rather than just correlation
- **Connectionism** - An approach to the study of human mental processes and cognition that utilizes mathematical models known as connectionist networks or artificial neural networks
- **Expert system** - A computer system emulating the decision-making ability of a human expert
- **Artificial general intelligence** - A hypothetical type of artificial intelligence that matches or surpasses human capabilities across virtually all cognitive tasks
- Core Concepts
 - **Embedding** - A representation learning technique that maps complex, high-dimensional data into a lower-dimensional vector space of numerical vectors
 - **Transfer learning** - A machine learning technique where knowledge gained from one task is reapplied to improve performance on a different but related task
 - **Mathematical model** - An abstract description of a concrete system using mathematical concepts and language
 - **Mathematical optimization** - The selection of a best element, with regard to some criteria, from some set of available alternatives
 - **Gradient descent** - A first-order iterative algorithm for minimizing a differentiable multivariate function
 - **Loss function** - A function that maps an event or values of one or more variables onto a real number intuitively representing some cost associated with the event
 - **Pattern recognition** - The task of assigning a class to an observation based on patterns extracted from data
 - **Feature engineering** - A preprocessing step in supervised machine learning and statistical modeling which transforms raw data into a more effective set of inputs

- **Overfitting** - The production of an analysis that corresponds too closely to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably
- **Bias-variance tradeoff** - The relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data
- **Inductive bias** - The set of assumptions that the learner uses to predict outputs of given inputs that it has not encountered
- **Curse of dimensionality** - The phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings

Machine Learning

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Machine Learning and Deep Learning

Learning Paradigms

- **Supervised learning** - A paradigm in machine learning where algorithms learn from labeled data
 - **Classification** - The problem of identifying which of a set of categories (sub-populations) a new observation belongs to, on the basis of a training set of data containing observations
 - **Logistic regression** - A statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables
 - **Support vector machine** - A set of supervised learning models with associated learning algorithms that analyze data for classification and regression analysis
 - **Naive Bayes classifier** - A family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features
 - **Decision tree learning** - A method using a decision tree as a predictive model to go from observations about an item to conclusions about its target value
 - **Ensemble learning** - A method using multiple learning algorithms to obtain better predictive performance than could be obtained from any of the

constituent learning algorithms alone

- **Random forest** - An ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time
- **ROC curve** - A graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied
- **Regression** - A set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables
 - **Ordinary least squares** - A type of linear least squares method for choosing the unknown parameters in a linear regression model
 - **Generalized linear model** - A flexible generalization of ordinary least squares regression
 - **ARIMA model** - A generalization of an autoregressive moving average (ARMA) model, fitted to time series data either to better understand the data or to predict future points in the series
- **Unsupervised learning** - A type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without previous training
 - **Principal component analysis** - A linear dimensionality reduction technique with applications in exploratory data analysis, visualization and data preprocessing
 - **K-means clustering** - A method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean
- **Reinforcement learning** - An area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward
 - **Markov chain** - A stochastic process that describes a sequence of events where the probability of each event depends only on the state attained in the previous event
 - **Markov decision process** - A mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker
 - **Hidden Markov model** - A statistical Markov model where the system being modeled is assumed to be a Markov process with unobserved (hidden) states
 - **Multi-armed bandit** - A problem in which a fixed limited set of resources must be allocated between competing (alternative) choices in a way that maximizes their expected gain

- [Value function](#) - A function used in mathematical optimization and reinforcement learning that assigns a measure of desirability to states or actions
- ML Concepts & Techniques
 - [Hyperparameter](#) - A parameter whose value is used to control the learning process
 - [Hyperparameter optimization](#) - The problem of choosing a set of optimal hyperparameters for a learning algorithm
 - [Early stopping](#) - A form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent
 - [Cross-validation](#) - Any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set
 - [Anomaly detection](#) - The identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data
 - [One-class classification](#) - The technique trying to identify objects of a specific class amongst all objects, by primarily learning from a training set containing only the objects of that class
 - [Recommender system](#) - An information filtering system that seeks to predict the 'rating' or 'preference' a user would give to an item
- ML Frameworks & Platforms
 - [scikit-learn](#) - A free software machine learning library for the Python programming language
 - [libsvm](#) - A Library for Support Vector Machines
 - [ML.NET](#) - An open-source, cross-platform machine learning framework for .NET developers
 - [Crab](#) - A Python library for building recommender systems
 - [mlxtend](#) - A Python library of useful tools for the day-to-day data science tasks
 - [Prophet](#) - A forecasting procedure for time series data that is fast and provides completely automated forecasts
 - [Azure Machine Learning](#) - An enterprise-grade machine learning service to build and deploy models faster
 - [Amazon SageMaker](#) - The service to build, train, and deploy machine learning (ML) models for any use case with fully managed infrastructure, tools, and workflows

- [Gradio](#) - The fastest way to demo your machine learning model with a friendly web interface so that anyone can use it, anywhere

Neural Networks and Deep Learning

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Machine Learning and Deep Learning
- Neural Network Fundamentals
 - [Neural network](#) - The computational models used in machine learning for finding patterns in data
 - [Tensor](#) - The mathematical objects represented as multidimensional arrays used in machine learning
 - Activation Functions
 - [Sigmoid function](#) - A mathematical function having a characteristic 'S'-shaped curve or sigmoid curve
 - [Softmax function](#) - A function that converts a vector of K real numbers into a probability distribution of K possible outcomes
 - [Backpropagation](#) - A widely used algorithm for training feedforward neural networks
 - [Autoencoder](#) - A type of artificial neural network used to learn efficient codings of unlabeled data (unsupervised learning)
 - [Vanishing gradient problem](#) - The difficulty encountered when training artificial neural networks with gradient-based learning methods and backpropagation, where gradients shrink as they back-propagate
- Deep Learning Concepts & Training
 - [Deep Learning](#) - A part of a broader family of machine learning methods based on artificial neural networks with representation learning
 - [Stochastic gradient descent](#) - An iterative method for optimizing an objective function with suitable smoothness properties
 - [Dropout \(neural networks\)](#) - A regularization technique for reducing overfitting in artificial neural networks by preventing complex co-adaptations on training data
 - [Fine tuning](#) - An approach to transfer learning in which the weights of a pre-trained model are trained on new data

- [LoRA \(machine learning\)](#) - A parameter-efficient fine-tuning technique for adapting pre-trained models to specific tasks with significantly fewer computational resources

Architectures

- [Recurrent neural network](#) - A class of artificial neural networks where connections between nodes can create cycles, allowing output from some nodes to affect subsequent input to the same nodes
 - [LSTM](#) - An artificial neural network used in the fields of artificial intelligence and deep learning, distinguished by feedback connections
- [Convolutional neural network \(CNN\)](#) - A class of artificial neural network, most commonly applied to analyze visual imagery
- [Attention](#) - A technique in the context of neural networks that mimics cognitive attention, enhancing the important parts of the input data and fading out the rest
 - [FlashAttention](#) - A fast and memory-efficient exact attention mechanism
 - [Transformer](#) - A deep learning architecture based on the multi-head attention mechanism
- [Mixture of experts](#) - A machine learning technique where multiple expert networks (learners) are used to divide a problem space into homogeneous regions
- DL Frameworks & Visualization
 - [TensorFlow](#) - An end-to-end open source platform for machine learning
 - [TFDS](#) - The collection of datasets ready to use with TensorFlow or other Python ML frameworks like Jax
 - [Keras](#) - The Python Deep Learning API designed for human beings, not machines
 - [PyTorch](#) - An open source machine learning framework that accelerates the path from research prototyping to production deployment
 - [ONNX Runtime](#) - A cross-platform, high-performance inference engine for machine learning models that accelerates both training and inference across diverse hardware and platforms
 - [Windows ML](#) - A unified, high-performance local AI inference framework for Windows powered by ONNX Runtime, enabling on-device model execution with hardware acceleration across NPUs, GPUs, and CPUs
 - [WinML CLI](#) - A command-line tool that converts and optimizes AI models for Windows ML, enabling hardware-optimized deployment across diverse Windows devices

- [AttentionViz](#) - A Global View of Transformer Attention
- [BertViz](#) - A tool for visualizing Attention in NLP Models
- Reference Books
 - [Neural Networks and Deep Learning](#) - A free online book explaining the core ideas behind neural networks and deep learning
 - [Deep Learning, MIT Press](#) - The textbook intended to help students and practitioners enter the field of machine learning in general and deep learning in particular

Natural Language Processing (NLP)

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Machine Learning and Deep Learning
- Linguistics Foundations
 - [Computational linguistics](#) - An interdisciplinary field concerned with the computational modelling of natural language, as well as the study of appropriate computational approaches to linguistic questions
 - [Morphology](#) - The study of words, how they are formed, and their relationship to other words in the same language
 - [Syntax](#) - A linguistic field that is the study of how words and morphemes combine to form larger units like phrases and sentences
 - [Semantics](#) - The study of linguistic meaning, examining how words acquire meaning and how complex expressions derive meaning from their constituent parts
 - [Distributional semantics](#) - A research area that develops and studies theories and methods for quantifying and categorizing semantic similarities between linguistic items based on their distributional properties in large samples of language data
 - [Symbol grounding problem](#) - The challenge of connecting abstract symbols to the real-world objects or concepts they represent

Concepts & Vector Representations

- [Okapi BM25](#) - A ranking function used by search engines to estimate the relevance of documents to a given search query based on a probabilistic retrieval framework

- [Levenshtein distance](#) - A string metric for measuring the difference between two sequences by counting the minimum number of single-character edits required to change one into the other
- [n-gram](#) - A sequence of n adjacent symbols in a particular order, used in fields like natural language processing and computational biology
- [tf-idf \(term frequency-inverse document frequency\)](#) - A statistical measure used in information retrieval to evaluate the importance of a word in a document relative to a collection or corpus, accounting for its general frequency
- [Word embedding](#) - A representation of a word in natural language processing, typically a real-valued vector that encodes its meaning such that words closer in vector space are expected to be similar in meaning
 - [Word2vec](#) - A technique in natural language processing for obtaining vector representations of words that capture information about their meaning based on surrounding words
 - [fastText](#) - Library for efficient text classification and representation learning
 - [GloVe](#) - Global Vectors for Word Representation
- [Sentence embedding](#) - A numerical vector representation of a sentence that encodes its meaningful semantic information
- NLP Libraries
 - [Natural Language Toolkit](#) - A leading platform for building Python programs to work with human language data
 - [Gensim](#) - A free open-source Python library for representing documents as semantic vectors
 - [Kuromoji](#) - An open source Japanese morphological analyzer written in Java
 - [Kagome](#) - An open source Japanese morphological analyzer written in pure golang
 - [mecab-python3](#) - A Python wrapper for the MeCab morphological analyzer for Japanese text
 - [jieba](#) - A Python module for Chinese text segmentation

Computer Vision

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Machine Learning and Deep Learning

Image Processing & Fundamentals

- [OpenCV](#) - An open source computer vision and machine learning software library
 - [GoCV](#) - A package for the Go programming language with bindings for OpenCV 4
- [Pillow](#) - A Python Imaging Library adding image processing capabilities
- [scikit-image](#) - A collection of algorithms for image processing in Python
- [ImageMagick](#) - A tool to create, edit, compose, or convert digital images

3D Vision & Geometry

- [Open3D](#) - An open-source library that supports rapid development of software that deals with 3D data
- [Point Cloud Library \(PCL\)](#) - A standalone, large scale, open source project for 2D/3D image and point cloud processing

Object Detection & Recognition

- [YOLO \(You Only Look Once\)](#) - An end-to-end object detection model designed for real-time detection
- [Detectron2](#) - Facebook's next generation library that provides state-of-the-art detection and segmentation algorithms
- [Mask R-CNN](#) - An extension of Faster R-CNN that adds a branch for predicting object masks in parallel with bounding boxes

Pose & Body Estimation

- [MediaPipe](#) - A framework for building multimodal ML pipelines for perception tasks like pose estimation, hand tracking, and face detection
- [OpenPose](#) - A library capable of real-time multi-person key point detection

Face & Biometric Recognition

- [face_recognition](#) - A Python library that recognizes and manipulates faces from Python or from the command line

OCR & Text Recognition

- [Tesseract OCR](#) - An open source text recognition (OCR) Engine
 - [gosseract OCR](#) - A Go package for OCR (Optical Character Recognition), by using

Tesseract C++ library

- [EasyOCR](#) - A ready-to-use OCR with 80+ supported languages and all popular writing scripts
- [OCRmyPDF](#) - A tool to add a searchable OCR text layer to PDF files
- Open Models
 - [LLaVA](#) - A novel end-to-end trained large multimodal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding, achieving impressive chat capabilities mimicking spirits of the multimodal GPT-4 and setting a new state-of-the-art accuracy on Science QA

Speech & Audio Processing

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.2 AI and Data Science > Machine Learning and Deep Learning

Automatic Speech Recognition (ASR)

- [Kaldi](#) - A speech recognition toolkit written in C++ for acoustic modeling and language modeling
- [DeepSpeech](#) - An open source engine for speech-to-text powered by deep learning
- [wav2vec 2.0](#) - A self-supervised learning approach for speech recognition pre-training
- Open Models
 - [Whisper](#) - A robust speech recognition model trained via large-scale weak supervision

Text-to-Speech (TTS) & Voice Synthesis

- [gTTS \(Google Text-to-Speech\)](#) - A Python library and CLI tool to interface with Google's Text-to-Speech API
- [Tacotron 2](#) - A neural network architecture for speech synthesis based on sequence-to-sequence models with attention

Audio Processing & Tools

- [librosa](#) - A Python library for audio and music analysis
- [PyAudio](#) - Python bindings for PortAudio, the cross-platform audio I/O library

- [SoX \(Sound eXchange\)](#) - The Swiss Army knife of sound processing programs
- [Praat](#) - A software program for the analysis of speech in phonetics

Voice Activity Detection & Audio Enhancement

- [webrtcvad](#) - A Python wrapper for the WebRTC voice activity detector
- [SileroVAD](#) - An open-source voice activity detector trained on large datasets

Speaker Identification & Voice Biometrics

- [Resemblyzer](#) - A Python package for speaker embedding-based speaker recognition
- [PyannoteAudio](#) - A Python toolkit for speaker diarization and speaker change detection

Generative AI & Large Language Models (LLMs)

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > AI Implementation and Operation
- Core Concepts
 - [Vision Language Models \(VLM\)](#) - An exciting class of models that can understand images and text
 - [Diffusion model](#) - A class of latent variable generative models in machine learning that learn to generate new data by reversing a gradual noising process
 - [Multimodal learning](#) - A deep learning approach that combines and processes diverse data types such as text, audio, images, or video for a more holistic understanding of complex information

Model Providers & Aggregators

- [Anthropic](#) - The API providing access to Anthropic's Claude models
- [OpenAI](#) - The platform for building applications with OpenAI's models
- [Gemini Developer APIs](#) - The API that gives you access to the latest Gemini models from Google
- [Hugging Face Serverless Inference API](#) - The API allowing inference on models hosted on the Hugging Face Hub
- [OpenRouter](#) - A unified interface for LLMs

- **Open Models**
 - [Llama](#) - The open-source AI models you can fine-tune, distill and deploy anywhere
 - [Gemma](#) - A family of lightweight, state-of-the-art open models built from the same research and technology used to create the Gemini models
 - [Mistral](#) - A family of open-source and commercial generative AI models
 - [OLMo](#) - A state-of-the-art, truly open language model and framework to build and study the science of language models
 - [Kimi K3](#) - An open-weight, native multimodal agentic model with 2.8 trillion parameters and a 1-million-token context window
 - [DeepSeek V4-Pro](#) - A Mixture-of-Experts (MoE) language model with 1.6T parameters (49B activated), a 1-million-token context window, and MIT-licensed open weights

Standards & Model Formats

- [GGUF](#) - A file format for storing models for inference with GGML and executors based on GGML
- [ONNX](#) - An open format built to represent machine learning models
- [Safetensors](#) - A simple format for storing tensors safely

Model Interface SDKs & Clients

- **Libraries & SDKs**
 - [Go OpenAI](#) - The Go client libraries for OpenAI API
 - [Google Gen AI SDK](#) - The Python SDK for Google's generative AI models
 - [Instructor](#) - A Python library designed to extract structured, validated data from Large Language Models (LLMs)
 - [OmniAI](#) - A minimalist library for interfacing with LLMs
 - [Outlines](#) - A library that guarantees structured outputs during generation directly from any large language model by enforcing compliance with JSON Schema, regular expressions, or context-free grammars
 - [Ruby OpenAI](#) - A Ruby wrapper for the OpenAI API
 - [RubyLLM](#) - The one beautiful Ruby API for GPT, Claude, Gemini, and more

Techniques & Methods

- [Retrieval-augmented generation \(RAG\)](#) - A technique that enables large language models to retrieve and incorporate new information from external data sources
 - [dsRAG](#) - A high-performance retrieval engine for unstructured data
- Model Compression
 - [Knowledge Distillation](#) - The process of transferring knowledge from a large machine learning model to a smaller one
 - [AWQ \(Activation-aware Weight Quantization\)](#) - An efficient and accurate low-bit weight quantization (INT3/4) for LLM compression and acceleration
 - [Pruning \(artificial neural network\)](#) - The practice of removing parameters from an existing artificial neural network
- [GraphRAG](#) - A data pipeline and transformation suite that is designed to extract meaningful, structured data from unstructured text using the power of LLMs
- [Prompt Engineering Guide](#) - A comprehensive resource for learning and applying prompt engineering techniques to effectively utilize large language models and build AI agents
- [CRAFT framework](#) - A structured method for crafting clear and precise AI prompts by defining context, role, action, format, and tone

Development Tools & Utilities

- [LiteLLM](#) - A Python SDK and AI Gateway (Proxy) that allows users to call over 100 Large Language Models (LLMs) using a unified OpenAI input/output format
- [RedCandle](#) - A Ruby gem for running state-of-the-art language models locally (via Rust's Candle)
- [Unsloth AI](#) - A platform providing tools and services for easily fine-tuning and training Large Language Models (LLMs) to achieve faster and more efficient AI training

Benchmarking & Analysis

- [ARC-AGI](#) - The benchmark measuring progress toward artificial general intelligence by testing skill-acquisition efficiency on novel tasks that are intuitive for humans but challenging for AI systems
- [OSWorld](#) - A scalable, real computer environment for evaluating multimodal agents on 369 open-ended tasks spanning web applications, desktop software, and cross-application workflows across Ubuntu, Windows, and macOS

- [FrontierMath](#) - An AI benchmark consisting of extremely challenging mathematical problems, including open research problems authored by expert mathematicians, ranging from undergraduate to research-level difficulty
- [MRCR v2](#) - A multi-round context recall benchmark that evaluates LLMs on their ability to retrieve and use information from extended multi-turn conversation histories
- [Artificial Analysis](#) - An independent analysis of AI models and API providers, helping users understand the AI landscape
- [Arena](#) - A platform designed for benchmarking and comparing various AI models, including both large language models (LLMs) and vision-language models (VLMs)

Agentic AI

Relevant DSS-P Skills

- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > Data / AI Utilization Strategy Design
- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > AI Implementation and Operation

Standards & Protocols

- [AGENTS.md](#) - An open standard for defining and running AI agents
- [Agent Skills](#) - A simple, open format for giving agents new capabilities and expertise
- [llms.txt](#) - A proposal to standardise on using an /llms.txt file to provide information to help LLMs use a website at inference time
- [Model Context Protocol \(MCP\)](#) - An open-source standard for connecting AI applications to external systems, enabling them to access data sources, tools, and workflows
- [A2A Protocol](#) - A protocol for enabling bidirectional communication between web applications and AI agents
- [Agent Name Service \(ANS\)](#) - A secure, DNS-inspired framework for AI agent discovery that leverages Public Key Infrastructure (PKI) for identity verification, structured JSON schemas for communication, and a protocol adapter layer supporting A2A, MCP, and ACP protocols
- [GitAgent](#) - A framework-agnostic standard that allows you to define an AI agent as a set of version-controlled files within a Git repository
- [AG-UI \(Agent-User Interaction Protocol\)](#) - An open, lightweight, event-based

protocol that standardizes how AI agents connect to user-facing applications

Agentic Patterns & Techniques

- [ReAct Prompting](#) - A prompting technique synergizing reasoning and acting in language models
 - Reason, Act, Thought, Observation
- [Recursive Language Models](#) - An inference strategy where language models (LMs) can decompose and recursively interact with input context of unbounded length
- [Effective Context Engineering for AI Agents](#) - The set of strategies for curating and maintaining the optimal set of tokens (information) during LLM inference, including all the other information that may land there outside of the prompts

Agent Orchestration & Frameworks

- Orchestration Frameworks
 - [Agno](#) - A multi-agent framework, runtime and control plane
 - [crewAI](#) - An open-source, multi-agent orchestration framework that empowers developers to orchestrate high-performing AI agents with ease and scale
 - [Deep Agents](#) - The easiest way to start building agents and applications powered by LLMs—with built-in capabilities for task planning, file systems for context management, subagent-spawning, and long-term memory
 - [Hermes Agent](#) - A self-improving AI agent with a built-in learning loop that creates skills from experience, improves them during use, maintains persistent memory, and builds a deepening model of users across sessions
 - [LangGraph](#) - A library for building stateful, multi-actor applications with LLMs by creating cyclic graphs for agent runtimes
 - [Mastra](#) - An all-in-one, open-source TypeScript framework for building, iterating, and deploying AI agents with built-in support for workflows, RAG, memory, and observability
 - [Microsoft Agent Framework](#) - A resource for building robust, future-proof Agentic AI solutions that evolve with technological advancements
 - [Microsoft 365 Agents SDK](#) - A comprehensive framework for building full-stack, multi-channel agents that operate seamlessly across Microsoft 365 Copilot, Teams, third-party platforms, custom applications, and websites
- Application Frameworks
 - [Chainlit](#) - An open-source Python package to build production ready

Conversational AI

- [DSPy](#) - A declarative framework for building modular AI software that allows for fast iteration on structured code and offers algorithms to compile AI programs into effective prompts and weights for language models
- [Genkit](#) - The AI framework for building full-stack applications with integrated support for agents, RAG, and tool use
- [LangChain](#) - A framework for developing applications powered by large language models
- [LlamaIndex](#) - A developer-first agent framework that rapidly accelerates time-to-production of GenAI applications with trusted low and high-level abstractions
 - [LlamaParse](#) - A document parser that turns even the most complex layouts, tables, charts, handwriting, checkboxes, and images into clean markdown
- [PydanticAI](#) - A Python agent framework for building production-grade applications with Generative AI, emphasizing type safety and structured outputs

Agent Development Kits (ADK)

- [Agent Package Manager \(APM\)](#) - An open-source dependency manager for AI agents that provides a single source of truth for skills, prompts, and instructions
- [Agent Development Kit \(ADK\)](#) - A flexible and modular framework for developing and deploying AI agents
- [Claude Agent SDK](#) - The Agent SDK gives you the same tools, agent loop, and context management that power Claude Code, programmable in Python and TypeScript
- [Claude Agent Skills](#) - The modular capabilities that extend an agent's functionality by packaging instructions, metadata, and optional resources
- [Fantasy](#) - A Go library for building AI agents with multiple providers and models through a single API
- [Go Micro](#) - An agent harness for Go that lets you build agents, services, and workflows on one runtime
- [Microsoft 365 Agents Toolkit](#) - A suite of tools for building enterprise-ready agents and apps that work across Microsoft 365 Copilot, Teams, Office, web, and other third-party messaging channels
- [OpenAI Agents SDK](#) - A library for building agentic applications where models can use additional context and tools, hand off to specialized agents, and stream results

Supporting Services & Platforms

- Agent Platforms & Services
 - [Foundry Agent Service](#) - A platform to securely design, deploy, and scale AI agents with governance and observability for enterprise transformation
 - [Moltbook](#) - A social network for AI agents where AI agents share, discuss, and upvote
- Interoperability
 - [FastMCP](#) - A Pythonic framework for building Model Context Protocol (MCP) servers and clients
- Pre-built Agents & Collections
 - [agency-agents](#) - A growing collection of meticulously crafted AI agent personalities designed to act as specialized experts with unique voices, proven workflows, and measurable deliverables
- Memory Systems
 - [Mem0](#) - An AI memory layer for LLM applications that aims to provide personalized AI experiences
 - [Graphiti](#) - An open-source Python framework for building temporally-aware context graphs
 - [Hindsight](#) - A memory system designed specifically for AI agents that enables them to retain, recall, and reason about information across multiple sessions using multi-strategy retrieval and automatic knowledge consolidation
- Search & Data Extraction
 - [Firecrawl](#) - An API service that takes a URL, crawls it, and converts it into clean markdown or structured data
 - [Tavily Search](#) - A search engine optimized for LLMs, aimed at efficient, quick and persistent search results
 - [SWIRL AI Search](#) - A Federated AI Search solution that connects to over 100 enterprise platforms, providing real-time visibility across data and document silos without requiring costly data transformations or migrations
- Security Tools
 - [skill-scanner](#) - A best-effort security scanner for AI Agent Skills that detects prompt injection, data exfiltration, and malicious code patterns
 - [Toxic Flow Analysis \(TFA\)](#) - The first principled approach to reduce the attack surface of AI applications, mitigating indirect prompt injections and other MCP attack vectors

MLOps & LLMOps

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > SRE Process
- 2. Data Preparation & Utilization > 2.1 Strategic Utilization of Data and AI > AI Implementation and Operation

ML Lifecycle & MLOps Platforms

- ML Lifecycle Tools
 - [DVC](#) - Open-source Data Version Control for machine learning projects
 - [CML](#) - An open-source tool for implementing continuous integration & delivery (CI/CD) in machine learning projects
 - [MLFlow](#) - An open source platform to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry
 - [KubeFlow](#) - The Machine Learning Toolkit for Kubernetes, dedicated to making deployments of ML workflows on Kubernetes simple, portable and scalable
 - [Weights & Biases](#) - The AI developer platform to build AI agents, applications, and models with confidence
 - [ZenML](#) - An open-source framework for orchestration, versioning, and governance from training pipelines to agent evals, local to Kubernetes
 - [BentoML](#) - An open-source framework to deploy any model anywhere, with tailored optimization, efficient scaling, and streamlined operations
- Managed MLOps Platforms
 - [Microsoft Foundry](#) - A unified, interoperable platform for building, optimizing, and governing AI apps and agents that understand business context and deliver business impact
 - [Vertex AI](#) - A machine learning (ML) platform for training and deploying ML models and AI applications
 - [Nebius](#) - A specialized AI cloud platform offering purpose-built infrastructure for AI and machine learning workloads
 - [Amazon SageMaker](#) - A fully managed service bringing together data, analytics, and AI to build, train, and deploy models at scale

LLM Serving & Runtimes

- Local LLM Runtimes
 - [LM Studio](#) - A desktop app for developing and experimenting with LLMs locally on your computer
 - [LocalAI](#) - The free, Open Source OpenAI alternative
 - [Ollama](#) - A tool designed for deploying and managing large language models (LLMs) locally
 - [llama.cpp](#) - A dependency-free C/C++ implementation for running LLM inference across hardware from Apple Silicon to NVIDIA GPUs
 - [Jan](#) - An open-source ChatGPT alternative that runs AI models locally on your computer
- Production Serving Engines
 - [vLLM](#) - A high-throughput and memory-efficient inference and serving engine for Large Language Models (LLMs)
 - [Hugging Face TGI](#) - A Rust, Python and gRPC server for text generation inference, used in production to power Hugging Chat and the Inference API
 - [SGLang](#) - A high-performance serving framework for large language models and multimodal models
 - [KServe](#) - A standardized distributed generative and predictive AI inference platform for scalable, multi-framework deployment on Kubernetes
 - [Modular MAX](#) - A unified AI inference platform delivering full optimizations from GPU kernel to API endpoint across diverse hardware

LLMOps & Observability

- Managed Model Services
 - [Amazon Bedrock](#) - A fully managed service offering a choice of high-performing foundation models
 - [Amazon Bedrock Agents](#) - A service that uses the reasoning of foundation models, APIs, and data to break down user requests, gather relevant information, and efficiently complete tasks
- Agent Observability
 - [Mission Control](#) - A centralized operational control plane to manage, monitor, and coordinate fleets of AI agents
- LLM Observability & Evals

- [Langfuse](#) - An open source LLM engineering platform providing traces, evals, prompt management, and metrics to debug and improve LLM applications
- [OpenLIT](#) - An open-source, OpenTelemetry-native tool for LLM and GenAI observability
- [LangSmith](#) - A unified DevOps platform for developing, collaborating, testing, deploying, and monitoring LLM applications
- [Helicone](#) - An open-source LLM observability platform built for developers to monitor and optimize their generative AI applications
- [Arize Phoenix](#) - An open-source AI observability and evaluation platform for LLMs
- [Braintrust](#) - The enterprise AI platform that provides an evaluation and observability platform for developers building with LLMs
- [Ragas](#) - A library that helps you move from vibe checks to systematic evaluation loops for your AI applications
- [DeepEval](#) - The LLM evaluation framework with research-backed metrics and pytest-native evaluations that run in CI/CD

07 - Fundamental Developer Skills

Software Development Methods

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Software Development Process
- 5. Personal Skill > 5.2 Conceptual Skill > Adaptability

Agile Development

- [Agile software development](#) - An umbrella term for approaches to developing software that reflect the values and principles agreed upon by The Agile Alliance
 - [Agile Manifesto](#) - A document that proclaims better ways of developing software by valuing individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan
 - [Software prototyping](#) - The activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed
 - [Minimum viable product](#) - A version of a product with just enough features to be usable by early customers who can then provide feedback for future product

development

- [User story](#) - An informal, natural language description of features of a software system
 - [Card, Conversation, Confirmation](#) - A practice where the Card is a token representing the requirement, the Conversation is where details are teased out, and the Confirmation is the acceptance test for the story
- [INVEST of PBI](#) - A mnemonic created by Bill Wake as a reminder of the characteristics of a good quality Product Backlog Item (PBI)
 - Independent: The PBI should be self-contained
 - Negotiable: Draft PBIs are not explicit contracts and should leave space for discussion
 - Valuable: A PBI must deliver value to the stakeholders
 - Estimable: You must always be able to estimate the size of a PBI
 - Small: PBIs should not be so big as to become impossible to plan/task/order within a level of accuracy
 - Testable: The PBI or its related description must provide the necessary information to make test development possible
- Key Methodologies
 - [Extreme Programming](#) - A software development methodology which is intended to improve software quality and responsiveness to changing customer requirements
 - [Scrum](#) - A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value
 - [Acceptance test-driven development](#) - A development methodology based on communication between the business customers, the developers, and the testers
 - [Three Amigos](#) - A meeting where the product owner, developer, and quality tester come together to establish clarity on the scope of the project
 - [Behavior driven development](#) - An agile software development process that encourages collaboration among developers, quality assurance testers, and customer representatives in a software project
 - [Specification by example](#) - A collaborative approach to defining requirements and business-oriented functional tests for software products based on capturing and illustrating requirements using realistic examples instead of abstract statements

- Key Practices
 - [Refactoring](#) - A disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior
 - [Software rot](#) - The tendency for software to deteriorate in quality, performance, or usefulness over time
 - [Technical debt](#) - A concept in software development that reflects the implied cost of additional rework caused by choosing an easy (limited) solution now instead of using a better approach that would take longer
 - [Technical Debt Ratio](#) - A metric used to measure the cost of fixing the existing technical debt in a codebase compared to the cost of developing the entire codebase from scratch
 - [Test driven development](#) - A software development process relying on software requirements being converted to test cases before software is fully developed, and tracking all software development by repeatedly testing the software against all test cases
- Tools for ATDD/BDD
 - [Gauge](#) - A free and open source test automation framework that takes the pain out of writing and maintaining acceptance tests
 - [Cucumber](#) - A tool that supports Behaviour-Driven Development (BDD) by reading executable specifications written in plain text and validating that the software does what those specifications say
 - [Gherkin Syntax](#) - A set of grammar rules that makes plain text structured enough for Cucumber to understand
 - [cucumber-ruby](#) - The Ruby implementation of Cucumber
 - [RSpec](#) - A testing tool for the Ruby programming language, created for behavior-driven development (BDD)
 - [Behave](#) - A tool for behavior-driven development (BDD) in Python, using tests written in a natural language style, backed up by Python code

Lean Development

- [Lean software development](#) - A translation of lean manufacturing principles and practices to the software development domain
 - [Continual improvement process](#) - An ongoing effort to improve products, services, or processes
 - [OODA loop](#) - A four-step approach to decision-making that focuses on filtering available information, putting it in context and quickly making the most

appropriate decision while also understanding that changes can be made as more data becomes available

- **Lean manufacturing** - A production method aimed primarily at reducing times within the production system as well as response times from suppliers and to customers
 - The 7 Wastes: The activities that add no value for the customer
 - **Value-stream mapping** - A lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from the beginning of the specific process until it reaches the customer
 - **ECRS (Eliminate, Combine, Rearrange, Simplify)** - A flexible continuous improvement strategy that allows for instant application of solutions to production lines
- **Toyota Production System** - An integrated socio-technical system, developed by Toyota, that comprises its management philosophy and practices
 - **Kanban** - A lean method to manage and improve work across human systems
 - **Kaizen** - A philosophy that focuses on continuous, incremental improvement of all functions and involves all employees from the CEO to the assembly line workers
 - **Autonomation** - The practice of designing equipment to partially automate a process and to automatically stop when a problem is detected, allowing operators to fix the issue immediately
 - **Heijunka** - A method for smoothing out production by leveling both the volume and the product mix
 - **Genchi Genbutsu** - A principle that suggests that to truly understand a situation, one needs to go to the "real place" where the work is done, observe the process, and verify the facts for themselves
 - **Andon (manufacturing)** - A visual management system used to indicate the status of a production line
 - **Muda (Japanese term)** - A key concept in lean process management as one of the three types of deviation from optimal allocation of resources, meaning futility, uselessness, or wastefulness
- **Theory of Constraints** - A management paradigm that views any manageable system as being limited in achieving more of its goals by a very small number of constraints

DevOps & Engineering Productivity

- Concepts
 - [CALMS framework](#) - A conceptual model for approaching DevOps that stands for Culture, Automation, Lean, Measurement, and Sharing
- Cultural & Organizational Foundations
 - [Generative organizational culture](#) - A type of culture characterized by a high degree of trust and cooperation, a shared sense of responsibility for the mission, and a focus on learning and continuous improvement
- Technical Practices
 - [Feature Toggles](#) - A powerful technique, allowing teams to modify system behavior without changing code
 - [Blue-Green Deployment](#) - A technique that reduces downtime and risk by running two identical production environments called Blue and Green
 - [Canary Release](#) - A technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure
 - [Everything as code](#) - A software development practice that seeks to apply the same principles of version control, testing, and deployment to enhance maintainability and scalability of all aspects of the development lifecycle, including networking infrastructure, documentation, and configuration

Release Automation

- [semantic-release](#) - A fully automated version management and package publishing tool that determines the next version number, generates release notes, and publishes the package based on formalized commit messages
- [Release Please](#) - A tool that automates changelog generation, the creation of GitHub releases, and version bumps for your projects based on Conventional Commits
- [GoReleaser](#) - A release automation tool for Go projects
- [Changesets](#) - A tool to manage versioning and changelogs with a focus on monorepos

Release Conventions & Standards

- [keep a changelog](#) - A file which contains a curated, chronologically ordered list of notable changes for each version of a project

- [Conventional Commits](#) - A lightweight convention on top of commit messages that provides an easy set of rules for creating an explicit commit history
- [Semantic Versioning](#) - A simple set of rules and requirements that dictate how version numbers are assigned and incremented
- [CalVer](#) - A versioning convention based on your project's release calendar, instead of arbitrary numbers

The Open Ecosystem

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development

Open Source

- [Open Source Initiative](#) - A non-profit organization dedicated to promoting and protecting open source software and communities
- Major Public Licenses
 - [MIT](#) - A permissive software license originating at the Massachusetts Institute of Technology (MIT) in the late 1980s
 - [BSD](#) - A family of permissive free software licenses, imposing minimal restrictions on the use and distribution of covered software
 - [Apache](#) - A permissive free software license written by the Apache Software Foundation (ASF), allowing users to use, distribute, and modify software without concern for royalties
 - [GPL](#) - A series of widely used free software licenses, or copyleft licenses, that guarantee end users the freedom to run, study, share, and modify the software
 - [LGPL](#) - A free-software license published by the Free Software Foundation (FSF) that allows developers and companies to use and integrate a software component released under it into their own (even proprietary) software without being required to release the source code of their own components
 - [AGPL](#) - A free, copyleft license for software and other kinds of works, specifically designed to ensure that if you run a modified program on a server and let other users communicate with it there, your server must also allow them to download the source code corresponding to the modified version running there
 - [SSPL](#) - A strong copyleft software license that mandates that any entity offering the licensed software as a service must release the complete source code of the entire service

- Principles & Adages
 - [Linus's law](#) - The assertion that 'given enough eyeballs, all bugs are shallow', which is a key principle in open source development
- Source Repositories
 - [GitHub](#) - The AI-powered developer platform to build, scale, and deliver secure software
 - [GitLab.com](#) - The DevSecOps Platform that empowers organizations to maximize the overall return on software development
- Package Registries
 - [CTAN](#) - The Comprehensive TEX Archive Network
 - [CPAN](#) - The Comprehensive Perl Archive Network
 - [CRAN](#) - The Comprehensive R Archive Network
 - [PyPI](#) - A repository of software for the Python programming language
 - [RubyGems.org](#) - The Ruby community's gem hosting service
 - [npm Registry](#) - The world's largest software registry
 - [JSR](#) - The open-source package registry for modern JavaScript and TypeScript
 - [pkg.go.dev](#) - Your source for information about Go packages and modules
 - [crates.io](#) - The Rust community's crate registry
 - [LuaRocks](#) - The package manager for Lua modules
 - [Hackage](#) - The Haskell community's central package archive of open source software
 - [Stackage](#) - A curated set of packages from Hackage
 - [NuGet Gallery](#) - The package manager for .NET
 - [Maven Central](#) - The World's Largest and Oldest Component Repository
 - [ConanCenter](#) - The central repository where you can find all the open source packages created by the community
 - [Anaconda Hub](#) - The Hub for Data Science and AI Collaboration

Open Data

- Tools and Licenses
 - [Creative Commons](#) - A nonprofit organization that helps overcome legal obstacles to the sharing of knowledge and creativity to address the world's pressing challenges

- [Open Data Commons](#) - The home of a set of legal tools and licenses to help you publish, provide and use open data
- Open Data Registries
 - [Hugging Face Hub](#) - A platform with over 900k models, 200k datasets, and 300k demos in which people can easily collaborate in their ML workflows
 - [Data.gov](#) - The home of the U.S. Government's open data
 - [Kaggle](#) - The world's largest data science community with powerful tools and resources to help you achieve your data science goals
 - [Registry of Open Data on AWS](#) - A service that makes it easy for people to find datasets that are publicly available through AWS services
 - [OpenML](#) - An open, collaborative, frictionless, and automated machine learning environment
 - [OpenStreetMap](#) - A map of the world, created by people like you and free to use under an open license
- Data Search Engines
 - [Google Dataset search](#) - A search engine that enables users to find datasets stored across thousands of repositories on the Web

Community & Governance

- Umbrella Open Source Foundations
 - [Linux Foundation](#) - A non-profit organization that supports, protects, and standardizes Linux by providing a neutral, trusted hub for developers to code, manage, and scale open technology projects
 - [Apache Software Foundation](#) - A nonprofit corporation to support Apache software projects, including the Apache HTTP Server
 - [Eclipse Foundation](#) - A provider of a business-friendly environment for open source software collaboration and innovation for a global community of individuals and organizations
- Technology-Specific Foundations
 - [OpenJS Foundation](#) - The neutral home for more than 40 open source projects, including Appium, Dojo, jQuery, Node.js, and webpack
 - [Rust Foundation](#) - An independent non-profit organization dedicated to stewarding and growing the Rust programming language and ecosystem
 - [Python Software Foundation](#) - The charitable organization behind the Python programming language

- [PyTorch Foundation](#) - A community-driven hub for open-source AI
- Cloud & AI
 - [Cloud Native Computing Foundation](#) - An open source software foundation dedicated to making cloud native computing universal and sustainable
 - [Agentic AI Foundation \(AAIF\)](#) - A neutral, open foundation to ensure this critical capability evolves transparently, collaboratively, and in ways that advance the adoption of leading open source AI projects
- Web & Data Standards
 - [World Wide Web Consortium](#) - An international community that develops open standards to ensure the long-term growth of the Web
 - [WHATWG](#) - A community of people interested in evolving HTML and related technologies
 - [The Open Group](#) - A global consortium that enables the achievement of business objectives through technology standards
- Ethical & Digital Rights
 - [Free Software Foundation](#) - A nonprofit with a worldwide mission to promote computer user freedom
- Community Governance & Codes of Conduct
 - [Debian Constitution](#) - A document that describes the structure of organisation for decision-making in the Debian Project
 - [Ubuntu Code of Conduct](#) - A set of guidelines that covers behavior as a member of the Ubuntu Community
 - [Mozilla Community Participation Guidelines](#) - A set of guidelines that outline expectations for participants within the Mozilla community
 - [Contributor Covenant](#) - A code of conduct for contributors to free/open source software projects, created by Coraline Ada Ehmke

Shell & Terminal

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- 3. Technology > 3.1 Software Development > Team Development

Bash & Other Shells

- **Bash** - An sh-compatible shell that incorporates useful features from the Korn shell (ksh) and the C shell (csh)
 - **Line editing** - The basic features of the GNU command line editing interface
 - **History** - The history expansion features of Bash
 - **Shell expansions** - The process performed on the command line after it has been split into words
 - **Pipelines** - A sequence of one or more commands separated by one of the control operators '|' or '&'
 - **Built-in commands** - The commands that are executed within the shell process itself, without forking a new process
 - **Special variables** - A list of shell variables that are set or used by the shell
 - **Built-in job control** - The ability to selectively stop (suspend) the execution of processes and continue (resume) their execution at a later time
- **Zsh** - A shell designed for interactive use, although it is also a powerful scripting language
- **fish-shell** - A smart and user-friendly command line shell for Linux, macOS, and the rest of the family
- **PowerShell** - A cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework
- **nushell** - A new type of shell

Shell Utilities

- General Shell Utilities
 - **coreutils** = A package of GNU software containing many of the basic tools, such as cat, ls, and rm, needed for Unix-like operating systems
 - **GNU parallel** - A shell tool for executing jobs in parallel using one or more computers
 - **rlwrap** - A readline wrapper
 - **bash-completion** - A collection of programmable completion functions for bash
 - **direnv** - An extension for your shell that can load and unload environment variables depending on the current directory
 - **zoxide** - A smarter cd command
 - **Atuin** - A tool that replaces your existing shell history with a SQLite database,

records additional context for your commands, and optionally syncs your shell history between all of your machines

- [wttr.in](#) - A console-oriented weather forecast service that delivers weather information through command-line tools like curl, httpie, or wget
- Search Tools
 - [findutils](#) - The basic directory searching utilities of the GNU operating system
 - [fzf](#) - A general-purpose command-line fuzzy finder
 - [fd](#) - A simple, fast and user-friendly alternative to find
 - [grep](#) - A command-line utility for searching plain-text data sets for lines that match a regular expression
 - [ripgrep](#) - A line-oriented search tool that recursively searches the current directory for a regex pattern
 - [silversearcher-ag](#) - A code-searching tool similar to ack, but faster
- Shell Frameworks & Customization
 - [starship](#) - The minimal, blazing-fast, and infinitely customizable prompt for any shell!
 - [oh-my-bash](#) - An open source, community-driven framework for managing your BASH configuration
 - [oh-my-zsh](#) - A delightful, open source, community-driven framework for managing your Zsh configuration
 - [Zim Framework](#) - The Zsh configuration framework with blazing speed and modular extensions
 - [Powerlevel10k](#) - A theme for Zsh
 - [Pure](#) - A pretty, minimal and fast ZSH prompt
- Shell Tutorials
 - [LinuxCommand.com](#) - A site containing a book and other material designed to help you learn how to use the Linux command line

Terminal Emulators

- [Terminal Emulators](#) - A computer program that emulates a video terminal within some other display architecture
 - [WaveTerm](#) - An open-source, cross-platform, AI-integrated terminal
 - [kitty](#) - The fast, feature-rich, GPU based terminal emulator
 - [Rio Terminal](#) - A modern terminal for the 21st century

- [Alacritty](#) - A modern terminal emulator that comes with sensible defaults, but allows for extensive configuration
- [Terminator](#) - A terminal emulator that puts multiple GNOME terminals in one window, letting you combine and recombine terminals to suit the style you like
- [Windows Terminal](#) - The new Windows Terminal and the original Windows console host
- [Mintty](#) - A terminal emulator for Cygwin, MSYS or Msys2, and derived projects, and for WSL
- [xterm](#) - A terminal emulator for the X Window System
- [wterm](#) - A terminal emulator for the web that renders directly to the DOM using a Zig+WASM core, enabling native browser text selection, copy/paste, and accessibility
- Technologies & Protocols
 - [Pseudoterminal](#) - A pair of pseudo-devices that provides a terminal-like interface used by programs to emulate a terminal
 - [ANSI escape code](#) - A standard for in-band signaling to control the cursor location, color, font styling, and other options on video text terminals
 - [kitty keyboard protocol](#) - A protocol for terminals to send keyboard events to applications running in them
 - [iTerm2 image protocol](#) - A custom escape code to display images inline in the terminal
- Terminal Fonts
 - [Nerd Fonts](#) - A project that patches developer targeted fonts with a high number of glyphs
 - [Share Tech Mono](#) - A monospaced sans serif, based on the Share family
 - [Cascadia Code](#) - A fun, new monospaced font that includes programming ligatures

Terminal Utilities

- Multiplexers & Session Management
 - [screen](#) - A full-screen window manager that multiplexes a physical terminal between several processes
 - [tmux](#) - A terminal multiplexer
 - [byobu](#) - A GPLv3 open source text-based window manager and terminal multiplexer

- [zellij](#) - A terminal workspace with batteries included
- [asciinema](#) - A free and open source solution for recording terminal sessions and sharing them on the web
- [ttyd](#) - A simple command-line tool for sharing terminal over the web
- Console File Managers
 - [midnight commander](#) - A visual file manager
 - [ranger](#) - A VIM-inspired filemanager for the console
 - [superfile](#) - A very fancy and modern terminal file manager

Linux or Unix-like environments on Windows

- [winpty](#) - A Windows software package providing a Unix-like VT100 console interface for Windows console programs
- [WSL](#) - A feature of Windows that enables you to run a GNU/Linux environment on your Windows machine without the need for a separate virtual machine or dual booting
 - [WSLg](#) - A project that enables support for running Linux GUI applications (X11 and Wayland) on Windows in a fully integrated desktop experience
- [Git for Windows](#) - A lightweight, native set of tools that bring the full feature set of the Git SCM to Windows
- [MSYS2](#) - A collection of tools and libraries providing you with an easy-to-use environment for building, installing and running native Windows software

Version Control System

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development
- Concepts
 - [Distributed Version Control](#) - A form of version control where the complete codebase, including its full history, is mirrored on every developer's computer
- Core VCS & Clients
 - [Git](#) - A free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency
 - local repository, remote repository
 - branch, tag, worktree

- push, pull, fetch, rebase, reset, stash
- staging, commit
- [Jujutsu \(jj\)](#) - A Git-compatible VCS that is both simple and powerful
- [gitoxide](#) - An idiomatic, lean, fast & safe pure Rust implementation of Git
- [TortoiseGit](#) - A Windows Shell Interface to Git and based on TortoiseSVN
- [git lfs](#) - An open source Git extension for versioning large files
- Terminal & UI Tools
 - [Informative git prompt for bash and fish](#) - A bash prompt that displays information about the current git repository
 - [lazygit](#) - A simple terminal UI for git commands
 - [gitui](#) - A tool that provides the comfort of a git GUI but right in your terminal
 - [giff](#) - A terminal-based Git diff viewer with interactive rebase capabilities that allows you to view and manage changes between branches
 - [Git Interactive Rebase Tool](#) - An improved sequence editor for Git
- History & Maintenance Tools
 - [BFG Repo-Cleaner](#) - A simpler, faster alternative to git-filter-branch for cleansing bad data out of your Git repository history
 - [git filter-repo](#) - A versatile tool for rewriting history
 - [degit](#) - Straightforward project scaffolding
- Commit & Changelog Tooling
 - [Git Lint](#) - A command line interface for linting Git commits by ensuring you maintain a clean, easy to read, debuggable, and maintainable project history
 - [commitlint](#) - A tool that helps your team adhere to a commit convention by linting commit messages
 - [git cliff](#) - A highly customizable changelog generator
- Hook Management
 - [pre-commit](#) - A framework for managing and maintaining multi-language pre-commit hooks
 - [Lefthook](#) - A fast, polyglot Git hooks manager for any type of project
 - [Husky](#) - Ultra-fast modern native git hooks
 - [lint-staged](#) - A tool that runs tasks like formatters and linters against staged git files

Git Hosting Services

- [GitLab SCM](#) - The single source of truth for collaborating on code and projects
 - [GitLab CLI](#) - An open source tool that brings GitLab to your terminal, next to where you are already working with git and your code
- [Gitea](#) - A painless self-hosted all-in-one software development service, including Git hosting, code review, team collaboration, package registry and CI/CD
- [Codeberg](#) - A community-led effort that provides Git hosting and other services for free and open source projects
- [Forgejo](#) - A self-hosted lightweight software forge
- [Soft Serve](#) - A tasty, self-hostable Git server for the command line
- [Azure Repos](#) - A set of version control tools that you can use to manage your code
- [GitHub](#) - The AI-powered developer platform to build, scale, and deliver secure software
 - [GitHub CLI](#) - An open source tool that brings pull requests, issues, GitHub Actions, and other GitHub features to your terminal, so you can do all your work in one place

Branching Models

- [Git Flow](#) - A branching model built around project releases, using long-lived develop and main branches alongside feature, release, and hotfix branches to keep parallel development organized
- [Trunk Based Development](#) - A source-control branching model, where developers collaborate on code in a single branch called 'trunk', resist any pressure to create other long-lived development branches by employing documented techniques
- [GitHub Flow](#) - A lightweight, branch-based workflow designed for teams that deploy frequently
- [GitLab Flow](#) - A simpler alternative to GitFlow that combines feature driven development and feature branches with issue tracking

Code Review

- [Google Engineering Practices Documentation](#) - A comprehensive guide to Google's code review processes and policies for both reviewers and change authors
- [Conventional Comments](#) - A standard that provides a structured format for code review feedback to improve clarity, reduce misunderstandings, and make comments machine-readable

- [Danger](#) - A tool to automate team norms in code review
- [Gerrit](#) - The open source, patchset-based code review platform for Git-based development

Integrated Development Environment (IDE)

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development

- GUI-based
 - [Eclipse IDE](#) - A free and open source IDE that is industry-backed and community driven, offering proven extensibility through a wide variety of platform plugins
 - [Visual Studio Code](#) - A lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux
 - [Awesome VS Code](#) - A curated list of delightful VS Code packages and resources
 - [GitLens](#) - An extension that supercharges the Git capabilities built into Visual Studio Code
 - [Git Graph](#) - An extension to view a Git Graph of your repository, and perform Git actions from the graph
 - AI Assistance Plugins
 - [GitHub Copilot](#) - The AI pair programmer that helps you write code faster and with less work
 - [Awesome GitHub Copilot](#) - A community-contributed collection of agents, instructions, skills, hooks, workflows, plugins, canvas extensions, and tools to enhance the experience
 - [Gemini Code Assist](#) - An AI-powered assistant for the entire development lifecycle
 - [Amazon Q Developer](#) - The most capable generative AI-powered assistant for software development
 - [Cline](#) - An open source AI coding agent that brings frontier AI models directly to your VS Code editor
 - AI-integrated IDEs
 - [Cursor](#) - An AI-native fork of VS Code designed for seamless, agentic integration with proprietary and frontier models
 - [Windsurf](#) - Where the work of developers and AI truly flow together, allowing

for a coding experience that feels like literal magic

- [Zed](#) - A next-generation code editor designed for high-performance collaboration with humans and AI
- Web-based
 - [code-server](#) - A VS Code instance running on a remote server, accessible through any web browser
 - [GitHub Codespaces](#) - A fully configured, secure cloud development environment native to GitHub that gets you up and coding faster
 - [Replit](#) - A platform to turn ideas into apps in minutes — no coding needed
- Terminal-based
 - [Vim](#) - A highly configurable text editor built to make creating and changing any kind of text very efficient
 - [motion and operators](#) - The commands that move the cursor and the commands used to delete or change text
 - [vim-plug](#) - The de-facto standard plugin manager for Vim
 - [NERDTree](#) - A tree explorer plugin for vim
 - [Neovim](#) - Hyperextensible Vim-based text editor
 - [LazyVim](#) - A Neovim setup powered by [l^z lazy.nvim](#) to make it easy to customize and extend your config
 - [lazy.nvim](#) - A modern plugin manager for Neovim
 - [NvChad](#) - A blazing fast Neovim config providing solid defaults and a beautiful UI
 - [neo-tree.nvim](#) - A Neovim plugin to manage the file system and other tree like structures
 - [colorful-winsep.nvim](#) - A colorful window separator for Neovim
 - [mason.nvim](#) - A Neovim plugin that allows you to easily manage external editor tooling such as LSP servers, DAP servers, linters, and formatters through a single interface
 - [telescope.nvim](#) - A highly extendable fuzzy finder over lists
 - [flash.nvim](#) - A plugin that helps you navigate your code with search labels, enhanced character motions and Treesitter integration
 - [nvim-llama](#) - A simple interface to Ollama for Neovim
 - [Helix](#) - A modal editor, meaning it has different modes for different tasks
 - [GNU Emacs](#) - An extensible, customizable, free/libre text editor — and more

- [MELPA](#) - Milkypostman's Emacs Lisp Package Archive
- [doomemacs](#) - An Emacs framework for the stubborn martian hacker
- [neotree](#) - A tree explorer for Emacs
- [Treemacs](#) - A tree layout file explorer for Emacs
- [Spacemacs](#) - A community-driven Emacs distribution
- Tutorials and Cheat Sheets
 - [OpenVim](#) - An interactive Vim tutorial
 - [Vim Adventures](#) - An online game based on VIM's keyboard shortcuts
 - [Vim Cheat Sheet](#) - A quick reference guide for Vim commands

Language Servers

- [LSP](#) - The protocol used between an editor or IDE and a language server that provides language features like auto complete, go to definition, and find all references
- [pyright](#) - A static type checker and language server for Python
 - [Pylance](#) - An extension that works alongside the Python extension in Visual Studio Code to provide performant language support
- [Ruby LSP](#) - An opinionated language server for Ruby
- [TypeScript Language Server](#) - A standalone TypeScript and JavaScript language server
- [Gopls](#) - The official language server for the Go language
- [rust-analyzer](#) - A language server for the Rust programming language
- [Eclipse JDT Language Server](#) - A Java language server based on the Eclipse JDT
- [clangd](#) - A language server that understands C++ code and adds smart features to your editor, including code completion, compile errors, and go-to-definition

Code Quality & Refactoring

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development
- Concepts
 - [SQALE method](#) - A method to support the evaluation of the quality of a software source code

- [Cyclomatic complexity](#) - A software metric used to indicate the complexity of a program
- [Code refactoring](#) - The process of restructuring existing source code without changing its external behavior
- [Code smell](#) - A characteristic of source code that hints at a deeper problem
- Code Metrics Tools
 - [scc](#) - A tool that counts lines of code in many programming languages
 - [cloc](#) - A tool that counts blank lines, comment lines, and physical lines of source code in many programming languages

Analysis Platform

- [SonarQube Server](#) - An on-premise analysis tool designed to detect coding issues in 30+ languages, frameworks, and IaC platform
- [CodeScene](#) - A code analysis platform that goes beyond static analysis, visualizing software quality through Code Health metrics and behavioral analysis of how teams work with code
- [Codecov](#) - A tool that measures code coverage and helps improve code quality at every step
- [GitLab Code Coverage](#) - A report that shows the percentage of your code that is covered by tests
- [GitLab Code Quality](#) - A feature that uses CodeClimate Engines to provide code quality analysis for your projects

Formatters

- [EditorConfig](#) - A file format for defining coding styles and a collection of text editor plugins that enable editors to read the file format and adhere to defined styles
- [Prettier](#) - An opinionated code formatter
- [Ruff](#) - An extremely fast Python linter and code formatter, written in Rust
- [Biome](#) - A toolchain of the web that provides a fast formatter and performant linter for JavaScript, TypeScript, JSX, JSON, HTML, CSS, and GraphQL
- [gofmt](#) - A tool that formats Go programs, using tabs for indentation and blanks for alignment
- [clang-format](#) - A tool to format C/C++/Java/JavaScript/JSON/Objective-C/Protobuf/C# code

Linters

- [ESLint](#) - An open source project that helps you find and fix problems with your JavaScript code
- [JSHint](#) - A Static Code Analysis Tool for JavaScript
- [Biome](#) - A toolchain of the web that provides a fast formatter and performant linter for JavaScript, TypeScript, JSX, JSON, HTML, CSS, and GraphQL
- [Pylint](#) - A static code analyser for Python 2 or 3
- [Ruff](#) - An extremely fast Python linter and code formatter, written in Rust
- [Staticcheck](#) - A state of the art linter for the Go programming language
- [revive](#) - Fast & extensible static code analysis framework for Go
- [golangci-lint](#) - A fast linters runner for Go
- [RuboCop](#) - A Ruby static code analyzer (a.k.a linter) and code formatter
- [Rust Clippy](#) - A collection of lints to catch common mistakes and improve your Rust code
- [PSScriptAnalyzer](#) - A static code checker for PowerShell modules and scripts
- [ShellCheck](#) - A GPLv3 tool that gives warnings and suggestions for bash/sh shell scripts
- [Stylelint](#) - A mighty CSS linter that helps you avoid errors and enforce conventions
- [vacuum](#) - An ultra-super-fast, lightweight OpenAPI linter and quality checking tool
- [yamllint](#) - A linter for YAML files
- [ls-lint](#) - An extremely fast file and directory name linter

Coding style guides

- [Google Style Guides](#) - A collection of documents that provide a set of conventions for writing source code in various programming languages
- [Style Guide for Python](#) - A document that gives coding conventions for the Python code comprising the standard library in the main Python distribution
- [Ruby Style Guide](#) - A community-driven style guide for the Ruby programming language
- [Airbnb JavaScript Style Guide](#) - A mostly reasonable approach to JavaScript

Developer AI & Productivity

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development

AI Productivity Tools

- [Mods](#) - A simple tool that helps you write programs with the assistance of AI
- [gptcli](#) - A command-line interface for ChatGPT
- [ShellGPT](#) - A command-line productivity tool powered by AI large language models (LLM)
- [LLM](#) - A CLI utility and Python library for interacting with Large Language Models
- [Fabric](#) - An open-source framework for augmenting humans using AI
- [OpenCommit](#) - Auto-generate meaningful commits in a second
- [AI Commits](#) - A CLI that writes your git commit messages for you with AI
- [lootbox](#) - A CLI which is inspired by "Code Mode" - LLMs write TypeScript code to call APIs rather than using tool invocation

Development Agents

- CLI Coding Agents
 - [Claude Code](#) - An agentic coding tool that reads your codebase, edits files, runs commands, and integrates with development tools
 - [Plan Mode](#) - A mode that instructs Claude to create a plan by analyzing the codebase with read-only operations, perfect for exploring codebases, planning complex changes, or reviewing code safely
 - [Hooks](#) - The user-defined shell commands or LLM prompts that execute automatically at specific points in the agent's lifecycle
 - [Subagents](#) - The specialized AI assistants that handle specific types of tasks, running in their own context with custom prompts and tool access
 - [Sandboxing](#) - A feature that provides filesystem and network isolation for safer, more autonomous agent execution, using OS-level primitives to enforce these isolations and reduce constant permission prompts
 - [Auto-memory](#) - A feature that allows the agent to automatically accumulate knowledge across sessions by saving notes on build commands, debugging insights, architecture notes, and user preferences without manual intervention
 - [Gemini CLI](#) - An open-source AI agent that brings the power of Gemini directly into your terminal
 - [Conductor](#) - The official project management tool for Gemini CLI

- [Sandboxing](#) - A feature that isolates potentially dangerous operations from your host system, providing a security barrier between AI operations and your environment, using macOS Seatbelt or container-based methods for isolation
- [Antigravity CLI](#) - A terminal-based AI coding agent that understands codebases, makes edits with permission, and executes commands directly from the terminal
- [GitHub Copilot CLI](#) - A tool that brings the power of Copilot coding agent directly to your terminal
- [Cursor CLI](#) - A tool built to help you ship right from your terminal, with the same commands in any environment
- [Amp](#) - The coding agent and development environment that runs anywhere and everywhere
- [Aider](#) - An AI pair programming tool in your terminal that lets you start a new project or build on your existing codebase
- [Letta Code](#) - A memory-first coding agent that lives in your terminal
- [Deep Agents CLI](#) - A terminal coding agent built on the Deep Agents SDK
- [OpenCode.ai](#) - An open-source AI coding agent for the terminal that provides an interactive TUI to help you understand, edit, and ship code within real repositories
- [OpenAI Codex](#) - A lightweight coding agent that runs in your terminal, providing a local coding assistant
- [Kimi Code](#) - A CLI coding agent by Moonshot AI engineered to drop into any dev workflow, supporting codebase analysis, file operations, web search, and parallel sub-agent task handling, powered by the kimi-k2.6 model
- [Crush](#) - The glamorous AI coding agent for your favourite terminal 🍷
- [ForgeCode](#) - A ZSH-integrated coding harness with a multi-agent architecture for research, planning, and execution, supporting multiple LLM providers and ranking at the top of Terminal-Bench
- Desktop Coding Agents
 - [goose](#) - A local AI agent that automates engineering tasks seamlessly
 - [Open Interpreter](#) - An open-source desktop agent that lets you work alongside agents that can read, edit, and create documents on your computer
 - [Agent Zero](#) - An open-source personal AI agent that uses a dynamic toolset to solve complex tasks by writing and executing code, managing its own computer environment, and learning from its actions

- [AionUI](#) - A Cowork app for AI coding agents with document generation capabilities
- Autonomous Coding Agents
 - [SWE-agent](#) - A tool for fixing bugs and issues in GitHub repositories, powered by a family of language models
 - [mini-swe-agent](#) - A smaller, more accessible version of SWE-agent
 - [Devin](#) - An autonomous AI software engineer that can handle complex engineering tasks unassisted within its own sandboxed environment
 - [Jules](#) - An Autonomous Coding Agent
 - [Replit Agent](#) - The first developer agent that can learn and work alongside you in your IDE

Supporting Tools & Infrastructure

- Platforms
 - [OpenHands](#) - A platform for software development agents powered by AI
 - [Port](#) - An agentic developer portal to build autonomous workflows to accelerate every aspect of engineering
 - [Antigravity](#) - An agentic development platform
 - [Warp](#) - The open platform for automating development
- Benchmarks
 - [SWE-bench](#) - A benchmark for evaluating large language models on real world software issues collected from GitHub
 - [Terminal-Bench](#) - A collection of harbor-native benchmark tasks for quantifying AI agents' terminal mastery, covering software engineering, machine learning, security, and data science
- Context Providers
 - [Context7](#) - An AI agent and tool designed to provide up-to-date documentation for LLMs and AI code editors
 - [LeanCTX](#) - An open-source context compression tool that reduces AI coding assistant token usage by compressing file reads and shell output by up to 99%, working with 29+ AI tools like Cursor and Claude Code
 - [Context Mode](#) - Context window optimization for AI coding agents. Sandboxes tool output, 98% reduction
 - [RTK](#) - A CLI proxy that reduces LLM token consumption by 60-90% on common

dev commands

- Semantic Code Retrieval
 - [Serena](#) - A powerful coding agent toolkit providing semantic retrieval and editing capabilities
- Session Tracking
 - [Entire](#) - A CLI tool that hooks into your git workflow to capture AI agent sessions on every push, creating a searchable record of how code was written and the intent behind every commit

08 - OS & Network Basics

Core OS Concepts

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [System call](#) - The programmatic way in which a computer program requests a service from the kernel of the operating system on which it is executed
- [Protection ring](#) - A mechanism to protect data and functionality from faults and malicious behavior
- [Daemon](#) - A computer program that runs as a background process, rather than being under the direct control of an interactive user
- [Environment variable](#) - A named variable whose value is set outside the program, typically through functionality built into the operating system or a microservice
- [POSIX standard](#) - A family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems

Process Management

- [Process](#) - The instance of a computer program that is being executed by one or more threads
 - [Thread](#) - The smallest sequence of programmed instructions that can be managed independently by a scheduler
 - [Scheduling](#) - The action of assigning resources to perform tasks
 - [Context switch](#) - The process of storing the state of a process or thread, so that it can be restored and resume execution at a later point
 - [Interrupt](#) - A request for the processor to interrupt currently executing code, so

that the event can be processed in a timely manner

Inter-Process Communication (IPC)

- Pipes
 - [Anonymous pipe](#) - A simplex FIFO communication channel that may be used for one-way interprocess communication
 - [Named pipe](#) - An extension to the traditional pipe concept on Unix and Unix-like systems, and is one of the methods of inter-process communication
- [Shared memory](#) - A memory that may be simultaneously accessed by multiple programs with an intent to provide communication among them or avoid redundant copies
- [Signal](#) - An asynchronous notification sent to a process or to a specific thread within the same process in order to notify it of an event that occurred
- [Unix domain socket](#) - A data communications endpoint for exchanging data between processes executing on the same host operating system

Memory Management

- [Virtual memory](#) - A memory management technique that provides an idealized abstraction of the storage resources that are actually available on a given machine
 - [Memory paging](#) - A memory management scheme by which a computer stores and retrieves data from secondary storage for use in main memory
 - [Page fault](#) - A type of exception raised by computer hardware when a running program accesses a memory page that is not currently mapped by the memory management unit into the virtual address space of a process
 - [Resident set size \(RSS\)](#) - The portion of memory occupied by a process that is held in main memory
 - [Working set size \(WSS\)](#) - The set of pages in the virtual address space of the process that are currently resident in main memory
- [Page cache](#) - A hardware or software component that stores data so that future requests for that data can be served faster

Storage Management

- [Disk partitioning](#) - The creation of one or more regions on a secondary storage device, so that each region can be managed separately
- [Loop device](#) - A pseudo-device that makes a file accessible as a block device

- **File system** - A method and data structure that the operating system uses to control how data is stored and retrieved
 - **Journaling file system** - A file system that keeps a journal, a circular log of changes that have not yet been committed to the main part of the file system
 - **Path** - The general form of the name of a file or directory, specifies a unique location in a file system
 - **Glob pattern** - A pattern that specifies sets of filenames with wildcard characters
 - **File handle/descriptor** - A unique identifier for a file or other input/output resource, such as a pipe or network socket
 - **Symbolic link** - A term for any file that contains a reference to another file or directory in the form of an absolute or relative path and that affects pathname resolution
 - **Permissions** - A feature of many modern file systems which control the ability of the users of a computer to view, change, navigate, and execute the contents of the file system
 - **Setuid** - A Unix access rights flag that allows users to run an executable with the permissions of the executable's owner or group
 - **Sticky bit** - A user ownership access right flag that can be assigned to files and directories on Unix-like systems
 - **Inode** - A data structure in a Unix-style file system that describes a file-system object such as a file or a directory
- **RAID** - A data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both

Base Network Concepts & Protocols

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- **The OSI Model** - A conceptual model that provides a common basis for the coordination of standards development for the purpose of systems interconnection

Link Layer (L2)

- **Ethernet** - A family of wired computer networking technologies
 - **ARP** - A communication protocol used for discovering the link layer address,

such as a MAC address, associated with a given internet layer address

- **MAC address** - A unique identifier assigned to a network interface controller for use as a network address in communications within a network segment
- **VLAN** - A broadcast domain that is partitioned and isolated in a computer network at the data link layer

Internet Layer (L3)

- **The Internet** - The global system of interconnected computer networks that uses the Internet protocol suite to communicate between networks and devices
- **IP** - The network layer communications protocol in the Internet protocol suite
 - **Link-local address** - A network address that is valid only for communications within the network segment or the broadcast domain that the host is connected to
 - **IP-multicast** - A method of sending Internet Protocol datagrams to a group of interested receivers in a single transmission
 - **IPv6** - The most recent version of the Internet Protocol, the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet
 - **Unique local address** - An IPv6 address in the address block fc00::/7
 - **ICMP** - A supporting protocol in the Internet protocol suite
 - **ICMPv6** - The implementation of the Internet Control Message Protocol for Internet Protocol version 6
 - **DHCP** - A network management protocol used on Internet Protocol networks for automatically assigning IP addresses and other communication parameters to devices connected to the network
 - **DHCPv6** - A network protocol for configuring Internet Protocol version 6 hosts with IP addresses, IP prefixes and other configuration data required to operate in an IPv6 network
 - **NAT** - A method of mapping an IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device
 - **NAT64** - An IPv6 transition mechanism that facilitates communication between IPv6 and IPv4 hosts
 - **NDP** - A protocol in the Internet protocol suite used with Internet Protocol Version 6
- Routing

- [Routing table](#) - A data table stored in a router or a network host that lists the routes to particular network destinations
- [CIDR](#) - A method for allocating IP addresses and for IP routing

Transport Layer (L4)

- [Network socket](#) - A software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network
- [TCP](#) - A main protocol of the Internet protocol suite
 - [TCP window scale option](#) - An option to increase the receive window size allowed in Transmission Control Protocol above its former maximum value of 65,535 bytes
- [UDP](#) - A core member of the Internet protocol suite
- [QUIC](#) - A UDP-based, stream-multiplexing, encrypted transport protocol

Network Architectures

- [Peer-to-peer](#) - A distributed computing or networking architecture in which participants share resources directly without reliance on a centralized administrative system

Domain Name System (DNS)

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [DNS](#) - The hierarchical and decentralized naming system used to identify computers, services, and other resources reachable through the Internet or other Internet Protocol networks
- [mDNS](#) - A protocol that resolves hostnames to IP addresses within small networks that do not include a local name server

Domain Registration & Lookup

- [IANA WHOIS Service](#) - A service to look up the registration data of a domain name or IP address
- [Registration Data Access Protocol \(RDAP\)](#) - A computer network communications protocol that delivers registration data from Domain Name Registries and Regional

Internet Registries

- [Domain drop catching](#) - The practice of registering a domain name once registration has lapsed, immediately after expiry

Server & Resolver Implementations

- [BIND \(dnstools\)](#) - A very flexible, full-featured DNS system
- [dnsmasq](#) - A lightweight, easy to configure DNS forwarder, DHCP and router advertisement server
- [CoreDNS](#) - A DNS server that chains plugins
- [systemd-resolved](#) - A system service that provides network name resolution to local applications
- mDNS Implementations
 - [Avahi](#) - A system which facilitates service discovery on a local network via the mDNS/DNS-SD protocol suite
 - [Bonjour](#) - Apple's implementation of zero-configuration networking

Client Tools

- Part of BIND
 - [dig](#) - A flexible tool for interrogating DNS name servers
 - [nslookup](#) - A program to query Internet domain name servers
- [dog](#) - A command-line DNS client
- [Doggo](#) - A modern command-line DNS client (like dig) written in Go
- Managed DNS Services
 - [Amazon Route53](#) - A highly available and scalable cloud Domain Name System web service
 - [Google Cloud DNS](#) - A high-performance, resilient, global Domain Name System service that publishes your domain names to the global DNS in a cost-effective way

Email System

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

- [Email](#) - A method of exchanging messages between people using electronic devices
- [SMTP](#) - A communication protocol for electronic mail transmission
- [POP](#) - An application-layer Internet standard protocol used by e-mail clients to retrieve e-mail from a mail server
- [IMAP](#) - An Internet standard protocol used by email clients to retrieve email messages from a mail server over a TCP/IP connection
- [MIME](#) - A standard that extends the format of email messages to support text in character sets other than ASCII
 - [Quoted-printable encoding](#) - An encoding that represents data in the 8-bit ASCII character set, so that it can be sent using a 7-bit data path
 - [Base64](#) - A group of binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation

Mailbox Formats

- Unix Mbox
- Maildir

Server Software (MTA/MDA)

- [Postfix](#) - A mail server that started life at IBM research as an alternative to the widely-used Sendmail program
- [Maddy Mail Server](#) - An all-in-one mail server that implements all functionality required to run a mail service
- Testing Servers
 - [Mailpit](#) - A small, fast, low memory, zero-dependency, multi-platform email testing tool & API for developers
- IMAP
 - [Cyrus IMAP](#) - A highly scalable enterprise mail system designed for use in small to large enterprise environments
 - [Dovecot](#) - An open source IMAP and POP3 email server for Linux/UNIX-like systems

Client Software & Utilities

- TUI Clients & Utilities
 - [mailutils](#) - A set of libraries and utilities for handling electronic mail

- [mail command](#) - A command to send and receive mail
- [Himalaya](#) - A CLI application for email management, built in Rust, that allows users to interact with emails through shell commands, with support for IMAP, Maildir, SMTP, OAuth 2.0, and PGP encryption
- [Mutt](#) - A small but very powerful text based program for reading and sending electronic mail under unix operating systems
- [swaks](#) - A featureful, flexible, scriptable, transaction-oriented SMTP test tool
- [msmtp](#) - An SMTP client that transmits mail to an SMTP server for delivery, designed to be configured as a sendmail replacement for mail clients
- [Pop](#) - A library for sending emails from your terminal
- [GNU sharutils](#) - A set of utilities for creating and unpacking shell archives
- Libraries
 - [go-mail](#) - A simple to use, yet feature rich mail library for Go
- GUI Clients
 - [Thunderbird](#) - A free email application that's easy to set up and customize
 - [Sylpheed](#) - A simple, lightweight but featureful, and easy-to-use e-mail client

Spam Test and Reputation

- [mail-tester](#) - A free online service that allows you to test your emails for Spam, Malformed Content and Mail Server Configuration problems
- [Spamhaus Project](#) - A non-profit organization that tracks spam and related cyber threats
- Cloud Services
 - [Amazon SES](#) - A cost-effective, flexible, and scalable email service that enables developers to send mail from within any application
 - [SendGrid](#) - A cloud-based email delivery platform that provides reliable transactional and marketing email delivery at scale

Unix-like Operating Systems

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [The Linux Kernel](#) - The main component of a Linux operating system and is the core interface between a computer's hardware and its processes

- Threads
 - [Pthreads](#) - An execution model that exists independently from a programming language, as well as a parallel execution model
- Filesystems
 - [Ext4](#) - The default file system for many major Linux distributions
 - [XFS](#) - A high-performance journaling file system created by Silicon Graphics, Inc
 - [Btrfs](#) - A copy-on-write filesystem for Linux with advanced features including snapshots, RAID, and self-healing
 - [UnionFS](#) - A filesystem service for Linux, FreeBSD and NetBSD which implements a union mount for other file systems
 - [OverlayFS](#) - A union mount filesystem implementation for Linux
 - [proc.5](#) - A virtual filesystem that provides an interface to kernel data structures
 - [sysfs.5](#) - A virtual filesystem that exports information about various kernel subsystems, hardware devices, and associated device drivers
- Container Support
 - [cgroups](#) - A Linux kernel feature which allow processes to be organized into hierarchical groups whose usage of various types of resources can then be limited and monitored
 - [namespaces](#) - A wrapper for a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource
 - [lxc/rootfs](#) - The userspace interface for the Linux kernel containment features
 - [nsenter](#) - A command that executes a program in the namespaces of other processes
- [FUSE \(Filesystem in Userspace\)](#) - An interface for userspace programs to export a filesystem to the Linux kernel
 - [s3fs](#) - A FUSE filesystem that allows you to mount an Amazon S3 bucket as a local filesystem
- [eBPF \(Extended Berkeley Packet Filter\)](#) - A revolutionary technology with origins in the Linux kernel that can run sandboxed programs in a privileged context

Linux Distributions

- General Purpose
 - [Arch Linux](#) - A simple, lightweight distribution
 - [Debian](#) - A complete Free Operating System
 - [Fedora](#) - An innovative, free and open-source operating system platform built by the community for hardware, clouds, and containers
 - [Gentoo](#) - A highly flexible, source-based Linux distribution that emphasizes customization and performance
 - [NixOS](#) - A Linux distribution with a unique approach to package and configuration management
 - [openSUSE](#) - A free Linux operating system for desktops, servers and containers
- Server-focused
 - [Ubuntu server](#) - The standard platform for public clouds, on-premises, and IoT devices
- Desktop-focused
 - Debian-based
 - [Ubuntu desktop](#) - The Linux-based operating system that runs from the desktop, to the cloud, to all your internet connected things
 - [BunsenLabs Linux](#) - A distribution offering a light-weight and easily customizable Openbox desktop
 - Arch-based
 - [Manjaro Linux](#) - A user-friendly Linux distribution based on the independently developed Arch operating system
 - [Mabox Linux](#) - Fast, lightweight and functional Linux Desktop "relaxed" rolling-release, Manjaro based with Openbox Window Manager

BSD Distributions

- [FreeBSD](#) - A free and open-source Unix-like operating system descended from the Berkeley Software Distribution (BSD)
- [NetBSD](#) - A free, fast, secure, and highly portable Unix-like Open Source operating system
- [OpenBSD](#) - A free and open-source, Unix-like operating system as a fork of NetBSD that emphasizes extremely high security and cryptography

System Services & Auth

- [Systemd](#) - A system and service manager for Linux operating systems
 - [journal](#) - A system service that collects and stores logging data
 - [hostname](#) - A system service that may be used to control the hostname and related machine metadata from user programs
 - [networkd](#) - A system service that manages networks
 - [resolved](#) - A system service that provides network name resolution to local applications
 - [timesyncd](#) - A system service that may be used to synchronize the local system clock with a remote Network Time Protocol server
- [linux-pam](#) - A system of libraries that handle the authentication tasks of applications and services in a Linux system

Machine Virtualization

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- [Virtualization](#) - The act of creating a virtual version of something, including virtual computer hardware platforms, storage devices, and computer network resources
- [libvirt](#) - A toolkit to manage virtualization platforms

Type-1 Hypervisors

- [KVM](#) - A full virtualization solution for Linux on x86 hardware containing virtualization extensions
- [Hyper-V](#) - A hardware virtualization product from Microsoft
- [Proxmox VE](#) - A complete, open-source server management platform for enterprise virtualization

Type-2 Hypervisors

- [VirtualBox](#) - A powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use
- [QEMU](#) - A generic and open source machine emulator and virtualizer

CPU Emulators

- [QEMU](#) - A generic and open source machine emulator and virtualizer

Computer Hardware

- Processing Units
 - [Central processing unit \(CPU\)](#) - The primary processor in a computer that executes program instructions by performing arithmetic, logic, controlling, and input/output operations
 - [Graphics processing unit \(GPU\)](#) - A specialized electronic circuit designed for accelerating graphics rendering and parallel computing tasks, widely used for 3D graphics, AI/ML workloads, and video processing
 - [Neural processing unit \(NPU\)](#) - A specialized hardware designed to accelerate artificial intelligence and machine learning applications through efficient execution of neural networks and computer vision tasks
- CPU Architectures
 - [x86-64](#) - A 64-bit version of the x86 instruction set
 - [ARM64](#) - The 64-bit extension of the ARM architecture family
- CPU Extensions
 - [x86 virtualization](#)
 - [Intel AMX](#)

Linux Host Administration

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

Core Utilities

- [util-linux](#) - A random collection of Linux utilities
 - [lsblk](#) - A command that lists information about all available or the specified block devices
 - [lsns](#) - A command that lists information about all the currently accessible namespaces or about the given namespace
 - [swapon](#) - A command used to specify devices on which paging and swapping are to take place

- [rsync](#) - An open source utility that provides fast incremental file transfer
- [sudo](#) - A system administrator to delegate authority to give certain users the ability to run some commands as root or another user
- [shadow-utils](#) - includes the necessary programs for converting UNIX password files to the shadow password format, plus programs for managing user and group accounts
 - [useradd](#) - A low level utility for adding users
- [jc](#) - A CLI tool and Python library that converts the output of popular command-line tools and file-types to JSON or Dictionaries
- [aha](#) - An Ansi HTML Adapter
 - [NO_COLOR](#) - An environment variable to disable ANSI color in command-line software
- [Vixie Cron](#) - An open source implementation of POSIX Cron
 - [Crontab.guru](#) - A quick and simple editor for cron schedule expressions
- [logrotate](#) - allows for the automatic rotation compression, removal and mailing of log files
- [Syslog](#) - A standard for message logging

Process & System Monitoring

- [procps](#) - A set of command line and full-screen utilities that provide information out of the pseudo-filesystem most commonly located at /proc
 - [ps](#) - A command that displays information about a selection of the active processes
 - [top](#) - A program that provides a dynamic real-time view of a running system
 - [free](#) - A command that displays the total amount of free and used physical and swap memory in the system
 - [vmstat](#) - A command that reports information about processes, memory, paging, block IO, traps, disks and cpu activity
- [psmisc](#) - A package of small utilities that use the proc file-system
 - [pstree](#) - A command that shows running processes as a tree
 - [killall](#) - A command that sends a signal to all processes running any of the specified commands
- [lsof](#) - A command for LiSting Open Files
- [strace](#) - A diagnostic, debugging and instructional userspace utility for Linux

- [inxi](#) - A full featured system information script
- [witr](#) - A tool that explains the causal ancestry and purpose of running processes
- Performance Monitors
 - [sysstat](#) - A collection of performance monitoring tools for Linux
 - [iostat](#) - A command used for monitoring system input/output device loading
 - [Monit](#) - A small Open Source utility for managing and monitoring Unix systems
 - [atop](#) - An ASCII full-screen performance monitor for Linux
 - [smem](#) - A tool that can give numerous reports on memory usage on Linux systems

Time Synchronization

- [NTP](#) - A networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks
- [chrony](#) - A versatile implementation of the Network Time Protocol
- [pool.ntp.org](#) - A big virtual cluster of timeservers providing reliable easy to use NTP service for millions of clients

Modern CLI Alternatives

- [gdu](#) - A fast disk usage analyzer with a console interface written in Go
- [lsd](#) - A rewrite of GNU ls with a lot of added features like colors, icons, tree-view, and more formatting options
- [eza](#) - A modern replacement for ls
- [broot](#) - A new way to see and navigate directory trees
- [bat](#) - A cat(1) clone with wings
- [dust](#) - A more intuitive version of du in rust
- [dua](#) - A tool to view disk space usage and delete unwanted data, fast
- [duf](#) - A better 'df' alternative
- [procs](#) - A modern replacement for ps written in Rust
- [htop](#) - An interactive process viewer for Unix systems
- [btop++](#) - A resource monitor for Linux, macOS, and FreeBSD
- [glances](#) - A cross-platform monitoring tool which aims to present a large amount of monitoring information through a curses or Web based interface
- [neofetch](#) - A command-line system information tool

Package Management Tools

- [dpkg](#) - The base package management system for Debian
 - [apt](#) - A command-line utility for installing, updating, removing, and otherwise managing deb packages on Ubuntu, Debian, and related Linux distributions
- [Pacman](#) - A utility which manages software packages in Linux
 - [Yay](#) - An AUR Helper Written in Go
- [yum](#) - An automatic updater and package installer/remover for rpm systems
- [dnf](#) - The next-generation version of yum
- [Homebrew](#) - The Missing Package Manager for macOS (or Linux)
- [pipx](#) - A tool to install and run Python applications in isolated environments
- [Flatpak](#) - A system for building, distributing, and running sandboxed desktop applications on Linux
- [Snapcraft](#) - A software packaging and deployment system developed by Canonical for operating systems that use the Linux kernel
- [arkade](#) - A portable marketplace for downloading your favourite devops CLIs and installing helm charts to your Kubernetes cluster
- [Plaza](#) - A customizable, riceable terminal UI for finding, installing, and managing packages across pacman, AUR, apt, dnf, and Flatpak

Linux Network Administration

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

Configuration & Management

- [iproute2](#) - A collection of utilities for controlling TCP / IP networking and traffic control in Linux
 - [ip](#) - The main command to show / manipulate routing, network devices, interfaces and tunnels
 - [ss](#) - A utility to investigate sockets
- [net-tools \(legacy\)](#) - A collection of programs for controlling the network subsystem of the Linux kernel
 - [ifconfig](#) - A command used to configure a network interface

- [netstat](#) - A command that prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
- [NetworkManager](#) - A daemon that sits on top of libudev and other Linux kernel interfaces and provides a high-level interface for network configuration
- [Ubuntu NetPlan](#) - A network configuration abstraction renderer

Analysis & Security

- [tcpdump](#) - A powerful command-line packet analyzer
- [wireshark](#) - The world's foremost network protocol analyzer
- [nmap](#) - An open source tool for network exploration and security auditing
 - [ncat](#) - A feature-packed networking utility which reads and writes data across networks from the command line
- [traceroute](#) - A computer network diagnostic tool for displaying the route and measuring transit delays of packets across an Internet Protocol network

Proxies & Gateways

- [SOCKS Proxy](#) - An Internet protocol that exchanges network packets between a client and server through a proxy server
 - [Dante](#) - A SOCKS server and SOCKS client, implementing RFC 1928 and related standards
 - [tun2socks](#) - A SOCKS proxy for TCP and UDP, that handles all connections from a TUN device
 - [proxychains](#) - A tool that forces any TCP connection made by any given application to follow through proxy like TOR or any other SOCKS4, SOCKS5 or HTTP(S) proxy
- Tunneling
 - [cloudflared](#) - The command-line client for Cloudflare Tunnel, a daemon that proxies traffic from the Cloudflare network to your origins
 - [ngrok](#) - A unified ingress platform that delivers reverse proxy, API gateway, and secure tunnels to expose local applications to the internet

File Sharing & Remote Access

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization

File Servers and Protocols

- **SMB** - A network communication protocol for providing shared access to files, printers, and serial ports between nodes on a network
 - **Samba** - The standard Windows interoperability suite of programs for Linux and Unix
- **FTP** - A standard communication protocol used for the transfer of computer files from a server to a client on a computer network
 - **vsftpd** - A GPL licensed FTP server for UNIX-like systems, including Linux
- **SFTP** - A network protocol that provides file access, file transfer, and file management over any reliable data stream
 - **SFTPGO** - A fully featured and highly configurable SFTP server with optional HTTP/S, FTP/S and WebDAV support

Remote Access Servers and Protocols

- **SSH** - A cryptographic network protocol for operating network services securely over an unsecured network
 - **openssh** - The premier connectivity tool for remote login with the SSH protocol
- **RDP** - A proprietary protocol developed by Microsoft which provides a user with a graphical interface to connect to another computer over a network connection
 - **xrdp** - An open-source Remote Desktop Protocol server
- **RFB** - A simple protocol for remote access to graphical user interfaces
 - **x11vnc** - A VNC server for X11
 - **TightVNC** - A free remote desktop application
- **Mosh** - A replacement for interactive SSH terminals
- Mesh VPN
 - **Tailscale** - A Zero Trust identity-based connectivity platform that replaces your legacy VPN, SASE, and PAM and connects remote teams, multi-cloud environments, CI/CD pipelines, Edge & IoT devices, and AI workloads

09 - Programming Concepts & Paradigms

Software Design & Architecture

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Software Design Methodology

Design Principles

- [Orthogonality and DRY principle](#) - The principle that every piece of knowledge must have a single, unambiguous, authoritative representation within a system
- [Separation of concerns](#) - A design principle for separating a computer program into distinct sections
- [Design by Contract](#) - An approach for designing software that prescribes formal, precise and verifiable interface specifications for software components
- [Law of Demeter](#) - A design guideline for developing software, particularly object-oriented programs
- [Hyrum's Law](#) - The law of implicit interfaces, stating that with a sufficient number of users of an API, all observable behaviors of your system will be depended on by somebody
- [SOLID - The principle of OOD](#) - A mnemonic acronym for five design principles intended to make object-oriented designs more understandable, flexible, and maintainable
 - Single responsibility
 - Open-closed
 - Liskov substitution
 - Interface segregation
 - Dependency inversion
- [The Reactive Manifesto](#) - A coherent approach to systems architecture where applications are responsive, resilient, elastic and message driven
- [Unix philosophy](#) - A set of cultural norms and philosophical approaches to software development
- [KISS principle](#) - A design principle which states that most systems work best if they are kept simple rather than made complicated
- [Wirth's law](#) - An adage on computer performance which states that software is getting slower more rapidly than hardware is becoming faster

Design Best Practices

- [Resource acquisition is initialization \(RAII\)](#) - A programming idiom where the life cycle of a resource is bound to the lifetime of an object
- [Rob Pike's 5 Rules of Programming](#) - A set of rules about where to focus

optimization efforts, emphasizing measurement and the importance of data structures

- [The Zen of Python](#) - A collection of 19 guiding principles for writing computer programs that influence the design of the Python programming language
- [The twelve-factor app](#) - A methodology for building software-as-a-service apps that are suitable for deployment on modern cloud platforms

Design Patterns

- [Software design pattern](#) - A general, reusable solution to a commonly occurring problem within a given context in software design
- [Entity-control-boundary](#) - An architectural pattern used in software design and analysis that helps in structuring the responsibilities of classes in an object-oriented system
- [Command Query Responsibility Segregation](#) - A pattern that separates read and update operations for a data store
- [Fluent interface](#) - A method for designing object-oriented APIs based on method chaining with the goal of making the readability of the source code close to that of ordinary written prose
- [Model-view-controller pattern](#) - A software design pattern commonly used for developing user interfaces that divides the related program logic into three interconnected elements
- [Dependency injection](#) - A design pattern in which an object or function receives other objects or functions that it depends on

Architectural Styles

- [Three-tier architecture](#) - A client-server architecture in which presentation, application processing, and data management functions are logically separated
- [Microservices architecture](#) - An approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms
- [Event-driven architecture](#) - A software architecture paradigm promoting the production, detection, consumption of, and reaction to events
- [Resource-oriented architecture](#) - A style of software architecture and programming paradigm for designing and developing software in the form of a network of resources
- [Background processing](#) - The execution of tasks in the background, allowing the

main application to remain responsive

Architecture Description

- **System** - A group of interacting or interrelated elements that act according to a set of rules to form a unified whole
 - **Systems architecture** - The conceptual model that defines the structure, behavior, and more views of a system
 - **4+1 architectural view model** - A view model used for "describing the architecture of software-intensive systems, based on the use of multiple, concurrent views"
 - **The C4 model** - An easy to learn, developer friendly approach to software architecture diagramming
 - **UML** - The graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
 - **Flowchart** - A type of diagram that represents a workflow or process
 - **Conway's law** - An adage stating that organizations design systems that mirror their own communication structure
- Related Standards
 - **ISO/IEC/IEEE 42010 (Architecture description)** - An international standard for architecture descriptions of systems and software

Domain-Driven Design (DDD)

- **Domain-driven design** - A major software design approach, focusing on modeling software to match a domain according to input from that domain's experts
 - **Ubiquitous Language** - The practice of building up a common, rigorous language between developers and users
 - **Anemic domain model** - A programming anti-pattern where the domain objects contain little or no business logic like validations, calculations, rules, and so forth
- **Object-oriented analysis and design** - A technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the software development process
 - **Use case** - A list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal

- **Ontology** - A representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many or all domains of discourse
 - **Semantic network** - A knowledge base that represents semantic relations between concepts in a network
 - **WordNet** - A large lexical database of English
- **Database design** - The organization of data according to a database model

Core Programming Concepts

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Language Mechanics & Execution

- **Source code** - A collection of code, possibly with comments, written using a human-readable programming language, usually as plain text
- **Statement** - A syntactic unit of an imperative programming language that expresses some action to be carried out
- **Expression** - A syntactic entity in a programming language that may be evaluated to determine its value
 - Operator and Operand
- **Literal** - A notation for representing a fixed value in source code
 - Template string or literal
 - **Heredoc** - A file literal or input stream literal representing a section of source code that is treated as if it were a separate file
- **Constant** - A value that cannot be altered by the program during normal execution
- **Variable** - An abstract storage location paired with an associated symbolic name, which contains some known or unknown quantity of information referred to as a value
- **Scope** - The region of a computer program where the binding of a name to an entity (name binding) is valid
- **Data type** - A collection or grouping of data values, usually specified by a set of possible values and allowed operations
 - **Primitives** - A data type provided by a programming language as a basic building block or one not defined in terms of other data types

- **Nominal type system** - A major class of type systems, in which compatibility and equivalence of data types is determined by explicit declarations and/or the names of the types
- **Structural type system** - A major class of type systems in which type compatibility and equivalence are determined by the type's actual structure or definition
- **Duck typing** - An application of the duck test determining type compatibility based on the presence of certain methods and properties
- **Union type** - A data type definition that specifies which of a number of permitted primitive types may be stored in its instances
- **Type variance** - The relationship between subtypes of a composite type and the subtypes of its components
- **Type safety** - The extent to which a programming language discourages or prevents type errors
- **Reference** - A value that enables a program to indirectly access a particular datum in the computer's memory or other storage device
 - **Null pointer** - A value saved for indicating that the pointer or reference does not refer to a valid object

Memory Management

- **Reference counting** - A programming technique of storing the number of references, pointers, or handles to a resource
- **Garbage collection** - A form of automatic memory management where the collector attempts to reclaim memory occupied by objects no longer in use
- **Smart pointer** - An abstract data type that simulates a pointer while providing added features, such as automatic memory management or bounds checking
- **Memory safety** - The state of being protected from various software bugs and security vulnerabilities when dealing with memory access

Control Flow Structures

- **Control flow** - The order in which individual statements, instructions or function calls of an imperative program are executed or evaluated
- **Continuation** - A data structure that represents the rest of a program's computation at a given point
 - **call-with-current-continuation** - A control flow operator, notably in the Scheme programming language, used to capture and invoke continuations

- **Exception handling** - The process of responding to the occurrence of exceptions during the execution of a program
- **Finite-state machine** - A mathematical model of computation representing an abstract machine that can be in exactly one of a finite number of states at any given time

Foundational Techniques & Properties

- **State** - The stored information, at a given instant in time, to which a computer program or system has access
- **Function** - A sequence of program instructions that performs a specific task, packaged as a unit
 - **Parameter** - A special kind of variable used in a subroutine or function to refer to one of the pieces of data provided as input
 - **Anonymous function** - A function definition that is not bound to an identifier
- **Immutable object** - An object whose state cannot be modified after it is created
- **Generic Programming** - A style of computer programming in which algorithms are written in terms of types to-be-specified-later that are then instantiated when needed
- **Assertion** - A statement that a predicate (a Boolean-valued function) is expected to always be true at that point in the code
- **Autovivification** - The automatic creation of a new variable or data structure as required when it is first used

Module Structure & Organization

- **Cohesion** - The degree to which the elements inside a module belong together
- **Coupling** - The degree of interdependence between software modules, a measure of how closely connected two routines or modules are, and the strength of the relationships between modules

Programming Paradigms

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Object-oriented Programming

- **Object-oriented Programming** - A programming paradigm based on the object - a software entity that encapsulates data and function(s)
 - **Abstraction** - The process of hiding the complexity of a system by modeling classes appropriate to the problem and working at the most relevant level of detail
 - **Encapsulation** - The bundling of data with the methods that operate on that data, or the restricting of direct access to some of an object's components
 - **Polymorphism** - The provision of a single interface to entities of different types
 - **Dynamic dispatch** - The process of selecting which implementation of a polymorphic operation (method or function) to call at run time
 - **Inheritance** - The mechanism of basing an object or class upon another object or class, retaining similar implementation
 - **Class** - An extensible program-code-template for creating objects, providing initial values for state and implementations of behavior
 - **Interface** - An abstract type that contains no data, but defines behaviors as method signatures
 - **Method** - A procedure associated with an object, and implicitly acting upon that object
 - **This keyword** - A keyword used in many object-oriented programming languages to refer to the object associated with the current function or method call
 - **Prototype-based programming** - A style of object-oriented programming in which behavior reuse is performed via a process of reusing existing objects that serve as prototypes

Functional Programming

- **Functional Programming** - A programming paradigm where programs are constructed by applying and composing functions
 - **Algebraic data type** - A composite data type formed by combining other types
 - **Pattern matching** - The act of checking a given sequence of tokens for the presence of the constituents of some pattern
 - **First-class function** - The property of a programming language that treats functions as first-class citizens (e.g., assignable to variables, passable as arguments)

- **Map** - A higher-order function that applies a given function to each element of a sequence, returning a sequence containing the results
- **Filter** - A higher-order function that processes a data structure to produce a new data structure containing only those elements for which a given predicate returns true
- **Reduce** - A higher-order function (also known as fold) that reduces a data structure to a single value by recursively applying a combining operation
- **Referential transparency** - A property of expressions such that an expression can be replaced with its corresponding value without changing the program's behavior
- **Closure** - A function together with a referencing environment for the non-local variables of that function
- **Side-effect** - An observable effect of an operation, function, or expression that modifies state variable values outside its local environment
- **Monad** - A software design pattern with a structure that combines program fragments (functions) and wraps their return values in a type with additional computation
- **Currying** - The technique of converting a function that takes multiple arguments into a sequence of functions that each takes a single argument

Reactive Programming & Others

- **Reactive Programming** - A declarative programming paradigm concerned with data streams and the propagation of change
 - **Functional Reactive Programming (FRP)** - A programming paradigm for reactive programming using the building blocks of functional programming
 - Languages & Frameworks
 - **RO** - A library for streams and reactive programming for Go
 - **ReactiveX** - An API for asynchronous programming with observable streams
 - **Elm** - A delightful language for reliable web applications
- **Aspect-oriented Programming** - A programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns
 - **Cross-cutting concern** - An aspect of a program that affect several modules, without the possibility of being encapsulated in any of them

Scripting Languages

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Python

- **Python** - A programming language that lets you work quickly and integrate systems more effectively
 - Core Features
 - **Python import system** - The mechanism that organizes Python code into modules and packages, facilitating code reuse and structuring large applications
 - **Special method names** - The methods, identified by leading and trailing double underscores, that allow classes to implement operations invoked by special syntax
 - **Type Hints** - A standard syntax for type annotations of variables, function parameters, and return values, used for static analysis
 - **typing module** - The standard library module providing runtime support for type hints
 - **Mypy** - An optional static type checker for Python that aims to combine the benefits of dynamic typing and static typing
 - **f-string** - A type of string literal, prefixed with 'f' or 'F', which allows embedding expressions inside string constants using minimal syntax
 - **with statement** - A statement that simplifies exception handling by encapsulating standard uses of try/finally statements for resource management
 - **contextlib** - A module that provides utilities for common tasks involving the with statement
 - **Generators** - A simple and powerful way to create iterators, defined using a function with the yield statement
 - **Decorators** - A syntax using the '@' symbol for transforming functions and methods, often used for modifying or enhancing them non-intrusively
 - **Coroutine** - A specialized generator function, defined with **async def**, that can suspend and resume its execution, enabling cooperative multitasking
 - **Lambda** - A small anonymous function defined using the **lambda** keyword,

restricted to a single expression

- [Data Classes](#) - A module and decorator providing a concise way to create classes primarily used to store data, automatically generating special methods
- [Pattern Matching](#) - A feature providing functionality similar to switch statements, allowing matching of values against complex patterns including sequences, mappings, and object structures
- [Unpacking Operator](#) - The extended usages of the [iterable unpacking operator](#) and [* dictionary unpacking operators](#) to allow unpacking in more positions, an arbitrary number of times, and in additional circumstances
- Key Libraries
 - [pathlib](#) - The module offering classes representing filesystem paths with semantics appropriate for different operating systems
 - [dotenv](#) - A library that reads key-value pairs from a `.env` file and can set them as environment variables
 - [Pydantic](#) - A data validation and settings management library for Python
 - [Tenacity](#) - A general-purpose retrying library for Python
- Development Tools
 - [IPython](#) - A rich interactive interface to Python with a history mechanism, tab completion, and special magic commands

JavaScript & TypeScript

- [JavaScript/ECMAScript](#) - The standard that defines the ECMAScript Language
 - Module System
 - [CommonJS](#) - A project with the goal of specifying an ecosystem for JavaScript outside the browser
 - [ES modules](#) - The official standard format to package JavaScript code for reuse
 - [UMD](#) - The patterns for Universal Module Definition for use in the browser, and in AMD and CommonJS-based systems
 - Core Features
 - [Event-driven](#) - A programming paradigm in which the flow of the program is determined by events such as user actions, sensor outputs, or messages from other programs
 - [Spread and rest operators](#) - The syntax that allows an iterable such as an

- array expression or string to be expanded in places where zero or more arguments or elements are expected
- [Generator](#) - An object returned by a generator function and it conforms to both the iterable protocol and the iterator protocol
- Key Libraries
 - [Lodash](#) - A modern JavaScript utility library delivering modularity, performance & extras
 - [dax](#) - Cross-platform shell tools for Deno and Node.js inspired by zx
 - [Bun Shell](#) - A built-in shell-like interface for running shell scripts
 - [zx](#) - A tool for writing better scripts
 - [Zod](#) - A TypeScript-first schema validation with static type inference
 - [yup](#) - A schema builder for runtime value parsing and validation
- [Typescript](#) - A strongly typed programming language that builds on JavaScript, giving you better tooling at any scale
 - [Union Types](#) - A way to combine multiple types into one
 - [Type Aliases](#) - A name for any type
 - [Type Assertions](#) - A way to tell the compiler 'trust me, I know what I'm doing'
 - [Mapped Types](#) - A generic type which uses a union of PropertyKeys to iterate through keys of another type to create a new one
 - [Nominal typing techniques](#) - A way to simulate nominal types in TypeScript, which by default has a structural type system
 - [Declaration Files](#) - The files where you define the types for a library
 - [Decorators](#) - A special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter
- TS Type Utilities
 - [json-schema-to-typescript](#) - A tool to compile JSONSchema to TypeScript type declarations
 - [Json Schema to TS](#) - The FromSchema method lets you infer TS types directly from JSON schemas
- Tutorials & Practices
 - [33 JS Concepts](#) - A repository with articles about 33 concepts every JavaScript developer should know
 - [JS Project Guidelines](#) - A set of best practices for JavaScript projects

- [Callback Hell](#) - The nesting of callback functions when dealing with asynchronous logic
- [NodeSchool](#) - A set of open source workshops that teach web software skills
- [Node.js Best Practices](#) - A summary and curation of the top-ranked content on Node.js best practices

Ruby, Perl & Others

- [Ruby](#) - A dynamic, open source programming language with a focus on simplicity and productivity
 - Core Features
 - [Percent notation](#) - A concise syntax for generating various literal types, such as strings, arrays, and regular expressions, using a percent sign and delimiters
 - [Fiber](#) - A lightweight concurrency primitive that allows for cooperative multitasking by pausing and resuming execution
 - [proc](#) - A mechanism to encapsulate a block of code into an object that can be stored, passed, and executed
 - [lambda](#) - A specialized block object that enforces strict argument checking and localized return behavior
 - [then](#) - A method that yields the object itself to a block and returns the result, facilitating functional-style method chaining
 - [define_method](#) - The ability to create and register methods at runtime using [define_method](#), enhancing code flexibility and reducing repetition
 - [instance_eval](#) - A method that evaluates a block or string within the context of a specific object instance, granting access to its internal scope and private methods
 - Libraries
 - [io-event](#) - The low level cross-platform primitives for constructing event loops
 - [Async](#) - A composable asynchronous I/O framework for Ruby based on io-event
 - [ruby-git](#) - A Ruby library that can be used to create, read and manipulate Git repositories by wrapping system calls to the git binary
 - Development Tools
 - [IRB \(Interactive Ruby\)](#) - A tool to interactively execute Ruby expressions read

from the standard input

- **Perl** - A family of two high-level, general-purpose, interpreted, dynamic programming languages
 - Core Features
 - **Special variables** - The variables that have a special meaning to Perl
 - **Built-in regex** - The syntax of regular expressions in Perl
 - **Context** - A property of expressions that determines how they behave when evaluated
 - **Scalar values** - A single item of data
 - **Reference** - A scalar data type that 'points' to another piece of data
 - **Quote-like operators** - A set of generic quoting operators
 - **I/O operators** - The operators used for input and output operations, such as reading from a filehandle
- **Tcl** - A dynamic programming language and a graphical user interface toolkit used for a wide range of applications
 - **Event-driven by design** - The built-in event loop that makes it ideal for GUIs and networking
- **Lua** - A powerful, efficient, lightweight, embeddable scripting language
- **Emacs Lisp** - The programming language used to extend and customize the Emacs text editor
 - **S-expression** - A notation for nested list (tree-structured) data
 - **Homoiconicity** - A property of some programming languages in which the primary representation of programs is also a data structure in a primitive type of the language itself

Asynchronous & Concurrency

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- **Concurrent computing** - A form of computing in which several computations are executed concurrently instead of sequentially
 - **Coroutine** - A computer program component that generalizes subroutines for non-preemptive multitasking, by allowing execution to be suspended and resumed

- [Async/await](#) - A syntactic feature that allows an asynchronous, non-blocking function to be structured in a way similar to an ordinary synchronous function
- [Futures and promises](#) - The constructs used for synchronizing program execution, representing a proxy for a result that is initially unknown
- [Semaphore](#) - A variable or abstract data type used to control access to a common resource by multiple threads in a concurrent system
- [Mutex](#) - A synchronization primitive that prevents state from being modified or accessed by multiple threads of execution at the same time
- [Channel](#) - A model for interprocess communication and synchronization via message passing
- [Thread safety](#) - A property of computer code applicable in multi-threaded environments, ensuring correct manipulation of shared data structures
- [Deadlock](#) - A situation in concurrent computing where no member of a group of entities can proceed because each waits for another member to take action

Language Analysis

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- Concepts
 - [Formal language](#) - A set of words, i.e. finite strings of letters, symbols, or tokens
 - [Well-formed formula](#) - A finite sequence of symbols from a given alphabet that is part of a formal language
 - [Formal grammar](#) - A set of formation rules for strings in a formal language
 - [Chomsky hierarchy](#) - A containment hierarchy of classes of formal grammars
 - [Rewriting](#) - A wide range of methods of replacing subterms of a formula with other terms
 - [Confluence](#) - A property of rewriting systems, describing which terms in such a system can be rewritten in more than one way, to yield the same result
 - [Automata theory](#) - The study of abstract machines and automata, as well as the computational problems that can be solved using them
 - [Finite-state machine](#) - A mathematical model of computation that can be in exactly one of a finite number of states at any given time, changing from one state to another in response to some inputs
 - [Lexical analysis](#) - The conversion of a text into meaningful lexical tokens

belonging to categories defined by a lexer program

- [Parsing](#) - A process of analyzing a string of symbols conforming to the rules of a formal grammar by breaking it into parts
- [BNF syntax](#) - A notation technique for context-free grammars, often used to describe the syntax of languages used in computing
- [AST](#) - A tree representation of the abstract syntactic structure of source code written in a programming language
- Parser Generators
 - [ANTLR](#) - A powerful parser generator for reading, processing, executing, or translating structured text or binary files
 - [Bison](#) - A general-purpose parser generator that converts a grammar description for a context-free grammar into a C program to parse that grammar
- Lexer Generators
 - [Flex](#) - The Fast Lexical Analyzer - scanner generator
 - [Ragel](#) - A state machine compiler
- Parsers/Libraries
 - [tree-sitter](#) - A parser generator tool and an incremental parsing library
 - [ts-morph](#) - A TypeScript Compiler API wrapper

Program Translation

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- Concepts
 - [Compiler](#) - A computer program that translates computer code written in one programming language into another language
 - [Transpiler](#) - A type of translator that takes the source code of a program written in a programming language as its input and produces an equivalent source code in the same or a different programming language
 - [Intermediate representation](#) - The data structure or code used internally by a compiler or virtual machine to represent source code
 - [Program optimization](#) - The process of modifying a software system to make some aspect of it work more efficiently or use fewer resources
 - [Machine code](#) - A computer program written in machine language instructions

that can be executed directly by a computer's central processing unit (CPU)

- [Cross compiler](#) - A compiler capable of creating executable code for a platform other than the one on which the compiler is running
- [Linker](#) - A computer system program that takes one or more object files and combines them into a single executable file

Major Compiler Infrastructures

- [LLVM Compiler Infrastructure](#) - A collection of modular and reusable compiler and toolchain technologies
 - [Clang](#) - A C language family frontend for LLVM
 - [LLD](#) - The LLVM Linker
- [gcc](#) - The GNU Compiler Collection which includes front ends for C, C++, Objective-C, Fortran, Ada, Go, and D
- [rustc](#) - The compiler for the Rust programming language

Specific Translators & Build Tools

- [MinGW-w64](#) - An advancement of the original mingw.org project, created to support the GCC compiler on Windows systems
- [Go build command](#) - A tool for managing Go source code
 - Static binary executable
- [GopherJS](#) - A compiler from Go to JavaScript
- [Bunster](#) - A shell compiler that turns your scripts into a self-contained executable programs

Linkers (Standalone)

- [gold](#) - A Modern Linker
- Runtime Libraries
 - [glibc](#) - The GNU C Library project which provides the core libraries for the GNU system and GNU/Linux systems
 - [musl libc](#) - A C standard library intended for operating systems based on the Linux kernel

Program Execution

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

- Concepts

- [Runtime System](#) - The part of a program that runs on a computer, for the language in which the program was written
- [Bytecode](#) - A form of instruction set designed for efficient execution by a software interpreter
- [Just-in-time compilation](#) - A way of executing computer code that involves compilation during execution of a program
- [Global interpreter lock](#) - A mutex that protects access to Python objects, preventing multiple threads from executing Python bytecodes at the same time

Runtime Implementations

- Javascript

- [Node.js](#) - A free, open-source, cross-platform JavaScript runtime environment
 - [libuv](#) - A multi-platform support library with a focus on asynchronous I/O
- [Deno](#) - A modern runtime for TypeScript and JavaScript
 - [Deno Deploy](#) - A distributed HTTP service that allows you to run JavaScript, TypeScript, and WebAssembly at the edge
 - [Deno Subhosting](#) - A platform for SaaS providers to securely run untrusted customer code at scale using V8 isolates
- [Bun](#) - A fast, all-in-one toolkit for running, building, testing, and debugging JavaScript and TypeScript
- [WinterJS](#) - A blazingly fast JavaScript runtime built on Rust, using the SpiderMonkey engine and the Tokio runtime

- Python

- CPython (default)
- [pypy](#) - A fast, compliant alternative implementation of Python
- [Pyodide](#) - A Python distribution for the browser and Node.js based on WebAssembly

- Ruby

- CRuby (default)
- [JRuby](#) - An implementation of the Ruby programming language atop the Java Virtual Machine

- [Java SE](#) - The most proven, trusted, and secure development platform for modern application development
 - [Java HotSpot VM](#) - The primary Java Virtual Machine for desktops and servers, produced by Oracle Corporation
 - [JMX API](#) - The Java Management Extensions technology which is a standard part of the Java Platform
 - [JDK tools](#) - The command-line tools to create and build applications
 - [GraalVM](#) - An advanced JDK with ahead-of-time Native Image compilation
 - [OpenJDK](#) - The place to collaborate on an open-source implementation of the Java Platform, Standard Edition
 - [Eclipse Temurin](#) - The open-source, enterprise-ready, and TCK-certified builds of OpenJDK
- [.NET](#) - The free, open-source, cross-platform framework for building modern apps and powerful cloud services
 - [CLR](#) - The virtual machine component of .NET Framework
- Runtime Utilities
 - [PM2](#) - A daemon process manager that will help you manage and keep your application online
 - [PyCall](#) - A Ruby library that allows you to call Python functions from Ruby
 - [VisualVM](#) - An All-in-One Java Troubleshooting Tool

Algorithm & Computational Complexity

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- Concepts
 - [Complexity class](#) - A set of computational problems of related resource-based complexity
- Reference Resources
 - [NIST Dictionary of Algorithms and Data Structures](#) - A dictionary of algorithms, algorithmic techniques, data structures, archetypal problems, and related definitions

Algorithm

- **Algorithm** - A finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation
 - Analysis Techniques
 - **Amortized analysis** - A method for analyzing a given algorithm's complexity
 - **Big O notation** - A mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity
 - Algorithmic Paradigms
 - **Recursion** - A method of solving a computational problem where the solution depends on solutions to smaller instances of the same problem
 - **Divide and conquer** - An algorithm design paradigm
 - **Dynamic programming** - A method for solving a complex problem by breaking it down into a collection of simpler subproblems
 - **Backtracking** - A class of algorithms for finding solutions to some computational problems
 - **Greedy algorithm** - An algorithmic paradigm that follows the problem-solving heuristic of making the locally optimal choice at each stage
 - Sorting Algorithms
 - **Quicksort** - An in-place sorting algorithm
 - **Merge sort** - An efficient, general-purpose, and comparison-based sorting algorithm
 - **Heapsort** - A comparison-based sorting algorithm
 - Searching Algorithms
 - **Binary search** - A search algorithm that finds the position of a target value within a sorted array
 - **Interpolation search** - An algorithm for searching for a key in a sorted array that has been ordered by numerical values assigned to the keys
 - String Algorithms
 - **Knuth–Morris–Pratt algorithm** - A string-searching algorithm that searches for occurrences of a "word" W within a main text string T
 - **Boyer–Moore algorithm** - A string-searching algorithm that is the standard benchmark for practical string-search literature
 - **Longest common subsequence** - The problem of finding the longest

subsequence common to all sequences in a set of sequences

- Graph Algorithms
 - Traversal
 - [Breadth-first search](#) - An algorithm for traversing or searching tree or graph data structures
 - [Depth-first search](#) - An algorithm for traversing or searching tree or graph data structures
 - Shortest Path
 - [Dijkstra's algorithm](#) - An algorithm for finding the shortest paths between nodes in a weighted graph
 - [Bellman-Ford algorithm](#) - An algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph
 - [Minimum Spanning Tree](#) - A subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together
 - [Prim's algorithm](#) - A greedy algorithm that finds a minimum spanning tree for a weighted undirected graph
 - [Kruskal's algorithm](#) - A minimum-spanning-tree algorithm which finds an edge of the least possible weight that connects any two trees in the forest
 - Other
 - [Tarjan's strongly connected components algorithm](#) - An algorithm in graph theory for finding the strongly connected components (SCCs) of a directed graph
 - [Topological sorting](#) - A linear ordering of the vertices of a directed acyclic graph (DAG)
- Hashing Algorithms
 - [Hash function](#) - Any function that can be used to map data of arbitrary size to fixed-size values

Data Structures

- [Abstract Data Types](#) - A mathematical model for data types
 - [String](#) - A finite sequence of symbols that are chosen from a set called an alphabet
 - [List](#) - An abstract data type that represents a finite number of ordered values

- **Associative array** - An abstract data type that can hold a collection of (key, value) pairs
- **Stack** - An abstract data type that serves as a collection of elements, with two main operations: push and pop
- **Queue** - An abstract data type that serves as a collection of elements, with two main operations: enqueue and dequeue
 - **Priority queue** - An abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it
- **Tree** - An abstract data type that represents a hierarchical tree structure with a set of connected nodes
- **Graph** - An abstract data type that is meant to implement the undirected graph and directed graph concepts from mathematics
 - **Directed acyclic graph (DAG)** - A directed graph with no directed cycles
- **Data Structures** - A data organization, management, and storage format that is designed to enable efficient access and modification
 - **Passive data structure** - A record data structure that contains only public data fields and provides no methods other than implicitly for reading/writing the fields
 - **Array** - A data structure consisting of a collection of elements (values or variables)
 - **Array slicing** - An operation that extracts a subset of elements from an array and packages them as another array
 - **Sparse matrix** - A matrix in which most of the elements are zero, allowing for specialized storage and algorithms to save memory and processing time
 - **Hash table** - A data structure that implements an associative array abstract data type
 - **Collision Resolution**
 - **Cuckoo hashing** - A scheme in computer programming for resolving hash collisions of keys in a hash table
 - **Linear probing** - A scheme in computer programming for resolving collisions in hash tables
 - **Linked data structure** - A data structure which consists of a set of data records (nodes) linked together and organized by references
 - **Persistent structure** - A data structure that always preserves the previous version of itself when it is modified

- [Disjoint-set data structure](#) - A data structure that stores a collection of disjoint (non-overlapping) sets
- Tree-based
 - [Search tree](#) - A tree data structure used for locating specific keys from within a set
 - [Binary search tree \(BST\)](#) - A rooted binary tree data structure with the key of each internal node being greater than all keys in the respective node's left subtree and less than the ones in its right subtree
 - [Merkle tree](#) - A tree in which every leaf node is labelled with the cryptographic hash of a data block
 - [Heap](#) - A tree-based data structure that satisfies the heap property
 - [Trie](#) - A search tree data structure used to locate specific keys from within a set
 - [Fenwick tree](#) - A data structure that can efficiently update elements and calculate prefix sums in a table of numbers
- Graph-based
 - [Adjacency matrix](#) - A square matrix used to represent a finite graph
 - [Adjacency list](#) - A collection of unordered lists used to represent a finite graph

10 - Advanced Programming

Procedural & Systems Programming

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Rust Language

- [Rust](#) - A programming language that empowers everyone to build reliable and efficient software
 - [Ownership and borrowing](#) - A set of rules that govern how a Rust program manages memory
 - [Interior mutability](#) - A design pattern in Rust that allows you to mutate data even when there are immutable references to that data
 - [Closure](#) - An anonymous function you can save in a variable or pass as an

argument to other functions

- [Trait-based generics](#) - A way to define behavior that a type must provide, allowing for generic code that can operate on any type that implements the specified behavior
- [Lifetime](#) - A construct the compiler uses to ensure all borrows are valid
- [Module Pin](#) - A module that provides types which pin data to its location in memory
- Tools
 - [C2Rust](#) - A tool that is able to translate most C modules into semantically equivalent Rust code
- Tutorials
 - [Rust by Example](#) - A collection of runnable examples that illustrate various Rust concepts and standard libraries

Go Language

- [Go](#) - An open-source programming language supported by Google
 - Core Features
 - [Go Modules](#) - The dependency management system for the Go programming language
 - [Defer, panic and recover](#) - The powerful but unusual control-flow mechanisms in Go
 - [Pointer receiver](#) - A method that operates on a pointer to the type, allowing it to modify the value to which the receiver points
 - [Interface](#) - A type defined as a set of method signatures
 - [Goroutine](#) - A lightweight thread managed by the Go runtime
 - [Channel](#) - A typed conduit through which you can send and receive values with the channel operator, `<`
 - Libraries
 - [Awesome Go](#) - A curated list of awesome Go frameworks, libraries, and software
 - [lo](#) - A Lodash-style Go library
 - [fp-go](#) - A collection of Functional Programming helpers
 - [shortuuid](#) - A generator library for concise, unambiguous and URL-safe UUIDs

- Tools
 - [Go binary size SVG treemap](#) - A CLI tool to make treemaps of size of Go executable
 - [mvm](#) - A fast virtual machine for Go and beyond that enables users to run Go programs directly from source code without compilation
- Tutorials
 - [Effective Go](#) - A document that gives tips for writing clear, idiomatic Go code
 - [Go by Example](#) - A hands-on introduction to Go using annotated example programs
 - [Learn Go with tests](#) - A resource that teaches the fundamentals of Go, including testing, on the first day

C & Other Procedural Languages

- [C](#) - A general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system
 - [Macros](#) - A fragment of code which has been given a name
- [Zig](#) - A general-purpose programming language and toolchain for maintaining robust, optimal and reusable software
 - [Comptime](#) - The mechanism that allows you to execute code at compile-time

Functional & Hybrid Programming

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Functional-first & Multi-paradigm Languages

- [F#](#) - A universal programming language for writing succinct, robust and performant code
 - Immutable data structure
 - [Discriminated union](#) - A type that can store a value of one of several different, but fixed, types
 - [Active pattern](#) - A feature that lets you define named partitions that subdivide input data, so that you can use these names in a pattern matching expression

- **Computation expression** - A feature that provides a convenient syntax for writing computations that can be sequenced and combined using control flow constructs and bindings
- **OCaml** - An industrial-strength functional programming language with an emphasis on expressiveness and safety
 - **Functor** - A module that is parameterized by another module
- **Haskell** - An advanced, purely functional programming language
 - Purely functional
 - **Lazy evaluation** - An evaluation strategy which delays the evaluation of an expression until its value is needed
- **Elixir** - A dynamic, functional language for building scalable and maintainable applications
 - **Process** - A lightweight thread of execution that is isolated and exchanges information via messages
- **Power Fx** - A low-code language used across Microsoft Power Platform that is a general-purpose, strong-typed, declarative, and functional programming language

Multi-paradigm & Hybrid Languages

- **Java** - The #1 programming language and development platform
 - **Built-in concurrency support** - The features of the Java platform designed from the ground up to support concurrent programming
- **C#** - A modern, object-oriented, and type-safe programming language
 - **Language-Integrated Query (LINQ)** - The name for a set of technologies based on the direct integration of query capabilities into the C# language
 - **Delegate** - A type that represents references to methods with a particular parameter list and return type
 - **Lambda expression** - A way to create an anonymous function
- **Scala** - A modern multi-paradigm programming language designed to express common programming patterns in a concise, elegant, and type-safe way
 - **Hybrid OO/functional** - A characteristic of a language that fuses object-oriented and functional programming in a statically typed setting
- **Groovy** - A powerful, optionally typed and dynamic language, with static-typing and static compilation capabilities, for the Java platform
- **Dart** - A client-optimized language for fast apps on any platform

Data & Format Standards

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Text Format & Character Code

- [ASCII](#) - A character encoding standard for electronic communication
- [Unicode](#) - The universal character encoding standard
 - [UTF-8](#) - A variable-width character encoding used for electronic communication
 - [Unicode Emoji](#) - A standardized set of characters that are used like emoticons
- [CSV](#) - A delimited text file that uses a comma to separate values
- [TSV](#) - A delimited text file format that uses a tab character to separate values in a table
- Libraries
 - [ICU](#) - A mature, widely used set of C/C++ and Java libraries providing Unicode and Globalization
 - [Python emoji](#) - An emoji library for Python
 - [Go emoji](#) - A minimalistic emoji package for Go

DateTime Format

- [UTC](#) - The primary time standard globally used to regulate clocks and time
- [ISO 8601](#) - An international standard covering the worldwide exchange and communication of date- and time-related data
- [Unix time](#) - A system for describing a point in time
- Libraries
 - [Ruby Time](#) - An abstraction of dates and times
 - [Python delorean](#) - A library for clearing up the inconvenient truths that arise dealing with datetimes in Python
 - [Python arrow](#) - A Python library that offers a sensible and human-friendly approach to creating, manipulating, formatting and converting dates, times and timestamps
 - [Luxon](#) - A powerful, modern, and friendly wrapper for JavaScript dates and times
 - [date-fns](#) - A modern JavaScript date utility library that provides the most

comprehensive, yet simple and consistent toolset for manipulating JavaScript dates in a browser & Node.js

- [Day.js](#) - A minimalist JavaScript library that parses, validates, manipulates, and displays dates and times for modern browsers with a largely Moment.js-compatible API
- [Go time](#) - A package that provides functionality for measuring and displaying time
- [Go when](#) - A natural language date/time parser with no dependencies
- [iCalendar](#) - A media type which allows users to store and exchange calendaring and scheduling information

Data Exchange Languages

- [JSON](#) - A lightweight data-interchange format
 - [jq](#) - A lightweight and flexible command-line JSON processor
 - [gojq](#) - A Pure Go implementation of jq
 - [gron](#) - A tool that transforms JSON into discrete assignments to make it easier to grep for what you want and see the absolute 'path' to it
 - [JMESPath](#) - A query language for JSON
 - [JSON::Tiny](#) - A minimalistic JSON module with no dependencies
 - [Python json](#) - A module that implements a JSON encoder and decoder
 - [Jackson](#) - A suite of data-processing tools for Java, including a streaming JSON parser / generator library and matching data-binding library
- [XML](#) - A simple, very flexible text format derived from SGML (ISO 8879)
 - [XPath](#) - An expression language that allows the processing of values conforming to the XQuery and XPath Data Model
 - [DOM](#) - A platform-neutral model for events, aborting activities, and node trees
 - [Python xml.etree.ElementTree](#) - A module that implements a simple and efficient API for parsing and creating XML data
 - [Nokogiri](#) - A Ruby library that makes it easy and painless to work with XML and HTML, providing tools for reading, writing, modifying, and querying documents
- [logfmt](#) - A log format that is simple, fast, and easy for humans and machines to parse
- [JSON Lines](#) - A convenient format for storing structured data that may be processed one record at a time

- Related Tools
 - [fx](#) - A terminal JSON viewer
 - [jnv](#) - An interactive JSON viewer and jq filter editor designed for navigating JSON

Configuration Languages

- JSON Superset
 - [Jsonnet](#) - A data templating language for app and tool developers
 - [Hjson](#) - A user interface for JSON
 - [YAML](#) - A human-friendly data serialization language for all programming languages
 - [yq \(python\)](#) - A command-line YAML, XML, TOML processor and jq wrapper for YAML, XML, TOML documents
 - [yq \(go\)](#) - A portable command-line YAML, JSON, XML, CSV, TOML and properties processor
 - [YAML::Tiny](#) - A Perl class for reading and writing YAML-style files, written with as little code as possible
 - [PyYAML](#) - A YAML parser and emitter for Python
 - [StrictYAML](#) - A type-safe YAML parser that parses and validates a restricted subset of the YAML specification
 - [JSON with comments](#) - A JS library to parse and stringify JSONC (JSON with comments)
 - [CUE](#) - An open-source data validation language and inference engine with its roots in logic programming
- Other Configuration Languages
 - [TOML](#) - A minimal configuration file format that's easy to read
 - [TOML::Tiny](#) - A minimal, pure perl TOML parser and serializer
 - [Python tomlib](#) - A module that provides an interface for parsing TOML
 - [HCL](#) - A toolkit for creating structured configuration languages that are both human- and machine-friendly
- Related Tools
 - [yj](#) - A command-line interface tool to convert between YAML, TOML, JSON, and HCL
- General Expression Languages

- [CEL](#) - A general-purpose expression language designed to be fast, portable, and safe to execute

Text Processing & Manipulation

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

Regular Expression

- [Regex](#) - A sequence of characters that specifies a search pattern in text
 - [PCRE](#) - A library implementing regular expression pattern matching using the same syntax and semantics as Perl 5
 - [Onigmo](#) - A regular expressions library forked from Oniguruma
 - [Python re](#) - The module provides regular expression matching operations similar to those found in Perl
 - [Go regexp](#) - The package that implements regular expression search
 - [RE2](#) - A fast, safe, thread-friendly alternative to backtracking regular expression engines
- [PRegex](#) - A Python library that allows for the programmatic creation of regular expressions
- [Regex Tools](#)
 - [Rubular](#) - A Ruby-based regular expression editor
 - [rubree](#) - A Ruby regular expression editor inspired by Rubular
 - [Wubular](#) - A Javascript-based regular expression editor, inspired by Rubular
 - [RegEx101](#) - An online regular expression editor and debugger

Text Tools

- [General Tools](#)
 - [GNU sed](#) - A stream editor used to perform basic text transformations on an input stream
 - [sd](#) - An intuitive find and replace command-line tool
 - [GNU diffutils](#) - A package of several programs for finding the differences between files
 - [colordiff](#) - A tool that produces the same output as diff but with coloured syntax

highlighting to improve readability

Tabular Data

- CLI Tools
 - [csvkit](#) - A suite of command-line tools for converting to and working with CSV
 - [xsv](#) - A fast CSV command line toolkit written in Rust
 - [qsv](#) - A command line program for indexing, slicing, analyzing, splitting, enriching, transforming & joining CSV files
 - [GNU awk](#) - A program that you can use to select particular records in a file and perform operations upon them
 - [GoAWK](#) - A POSIX-compliant AWK interpreter written in Go with CSV support
- Libraries
 - [Text::CSV](#) - A comma-separated values manipulator (using XS or PurePerl)
 - [Python csv](#) - A module that implements classes to read and write tabular data in CSV format
 - [Ruby csv](#) - A complete interface to CSV files and data
 - [smarter_csv](#) - A Ruby Gem for convenient reading and writing of CSV files
 - [Go csv](#) - A package that reads and writes comma-separated values (CSV) files
 - [Papa Parse](#) - The powerful, in-browser CSV parser for JavaScript
 - [Python tabulate](#) - A library and a command-line utility that displays data in a visually appealing format
 - [Text::MarkdownTable](#) - A module that can be used to write data in tabular form, formatted in MultiMarkdown syntax
 - [Terminal Table](#) - A simple, feature-rich ascii table generation library for ruby
 - [go-pretty](#) - A table-writer and more in golang

Template Engines

- Template Languages and Engines
 - [gomplate](#) - A fast template renderer supporting many datasources and hundreds of functions
 - [Go template](#) - A package that implements data-driven templates for generating textual output
 - [sprig](#) - A library that provides template functions for Go's template language

- [mustache](#) - A logic-less template syntax
- [Jinja](#) - A full-featured template engine for Python
- [Perl Text::Template](#) - A library for generating form letters, building HTML pages, or whatever you can imagine
- [Perl HTML::Template](#) - A system for creating HTML templates
- [Template Toolkit](#) - A fast, flexible and highly extensible template processing system
- [ERB](#) - An easy to use but powerful templating system for Ruby
- [Haml](#) - A markup language that's used to cleanly and simply describe the HTML of any web document without the use of inline code
- [Liquid](#) - A safe, customer-facing template language for flexible web apps
- [envsubst in gettext](#) - A program that substitutes the values of environment variables

Markup & Document Processing

- [unified](#) - A friendly interface backed by an ecosystem of plugins built for creating and manipulating content
 - [remark](#) - A markdown processor powered by plugins
- [markdown-it](#) - A Markdown parser with 100% CommonMark support, extensions, and syntax plugins
 - [markdown-it-py](#) - A Python port of the markdown-it project
- [markdownify](#) - A Python library that converts HTML to Markdown with customizable options for handling tags, formatting, and styling

Debugging

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- Debuggers
 - Python
 - [VSCode Python extension](#) - An extension with rich support for the Python language
 - [debugpy](#) - An implementation of the Debug Adapter Protocol for Python 3

- Node.js
 - [VSCode built-in debugger](#) - The built-in debugger that helps you speed up your edit, compile, and debug loop
 - [Node.js built-in inspector](#) - The inspector which allows attaching Chrome DevTools to Node.js instances for debugging and profiling
- Go
 - [VSCode Go extension](#) - An extension that provides rich language support for the Go programming language
 - [Delve](#) - A debugger for the Go programming language
- Ruby
 - [VSCode rdbg Ruby Debugger](#) - A Ruby debugger extension that is based on debug.gem
 - [debug.rb](#) - The debugging functionality for Ruby
- Others
 - [VSCode Bash Debug](#) - A bash debugger GUI frontend based on bashdb
 - [BASH Debugger](#) - A bash shell command-line debugger
 - [GDB](#) - The GNU Project debugger
- Debugger protocols
 - [DAP](#) - The abstract protocol used between a development tool (e.g. IDE or editor) and a debugger
 - [V8 V8 Inspector Protocol](#) - The protocol that allows for tools to instrument V8 to debug and profile JavaScript applications

Logging

- Logging Libraries
 - Python
 - [Python logging](#) - The module that defines functions and classes which implement a flexible event logging system for applications and libraries
 - [loguru](#) - A library which aims to bring enjoyable logging in Python
 - Javascript/Typescript
 - [bunyan](#) - A simple and fast JSON logging library for node.js services
 - [winston](#) - A logger for just about everything
 - [debug](#) - A tiny JavaScript debugging utility modelled after Node.js core's

debugging technique

- Go
 - [Charmbracelet log](#) - A minimal and colorful Go logging library
 - [Go log](#) - The package that implements a simple logging package
 - [zap](#) - Blazing fast, structured, leveled logging in Go
 - [Logrus](#) - A structured logger for Go (golang), completely API compatible with the standard library logger
 - [Zero Allocation JSON Logger](#) - The package that provides a fast and simple logger dedicated to JSON output
- Others
 - [logger](#) - A tool to enter messages into the system log
 - [log4j](#) - A versatile, industrial-grade Java logging framework
 - [log4sh](#) - An advanced logging framework for shell scripts
 - [log4net](#) - A port of the excellent Apache log4j framework to the Microsoft .NET runtime

Test Frameworks & Tools

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Software Development Process
- Test Concepts and Best Practices
 - [Test Pyramid](#) - A way of thinking about how different kinds of tests should be used to create a balanced portfolio
 - [Test case](#) - A specification of the inputs, execution conditions, testing procedure, and expected results that define a single test
 - [Test double](#) - An object that can stand in for a real object in a test
 - [Arrange-Act-Assert Pattern](#) - A pattern for structuring a test into setting up the object to be tested, acting on it through some mutator, and making claims about the object, its collaborators, and its parameters
 - [DAMP \(Descriptive And Meaningful Phrases\)](#) - A principle to lean more heavily towards, and less on DRY, when writing unit tests
 - [Unit testing best practices with .NET](#) - A set of best practices that help you write tests that are robust and easy to maintain
 - [JS Testing Best Practices](#) - A summary of the top testing practices for JavaScript

- Test Protocols
 - [Test Anything Protocol](#) - A simple text-based interface between testing modules and a test harness
 - [tappy](#) - A set of tools for working with the Test Anything Protocol (TAP)
 - [Node-Tap](#) - A Test-Anything-Protocol library for JavaScript

Test Frameworks

- Bash
 - [Bats-core](#) - A Bash Automated Testing System
 - [shUnit2](#) - A unit test framework for Bourne based shell scripts
 - [shellspec](#) - A full-featured BDD unit testing framework for dash, bash, ksh, zsh and all POSIX shells
- Ruby
 - [Minitest](#) - A complete suite of testing facilities supporting TDD, BDD, mocking, and benchmarking
 - [RSpec](#) - A testing tool for the Ruby programming language
 - [aruba](#) - A tool to test command-line applications with Cucumber-Ruby, RSpec or Minitest
- Python
 - [Python unittest](#) - A unit testing framework, sometimes referred to as 'PyUnit', which is a Python language version of JUnit
 - [pytest](#) - A framework that makes it easy to write small, readable tests, and can scale to support complex functional testing
 - [pytest-mock](#) - A pytest plugin that provides a [mock](#) fixture which is a thin-wrapper around the standard [unittest.mock](#) package
- Javascript/Typescript
 - [Vitest](#) - A blazing fast unit-test framework powered by Vite
 - [Jest](#) - A delightful JavaScript Testing Framework with a focus on simplicity
 - [Mocha](#) - A feature-rich JavaScript test framework running on Node.js and in the browser
 - Runtime-integrated
 - [bun test](#) - A fast, Jest-compatible test runner built into Bun
 - [deno test](#) - A built-in test runner that you can use for testing JavaScript and

TypeScript code

- Go
 - [Go testing](#) - A package that provides support for automated testing of Go packages
 - [Ginkgo](#) - A BDD-style testing framework for Go
- Others
 - [JUnit](#) - The 5th major version of the programmer-friendly testing framework for Java and the JVM
 - [xUnit.net](#) - A free, open source, community-focused unit testing tool for the .NET Framework

Assertion Libraries

- [Chai](#) - A BDD / TDD assertion library for node and the browser
- [Gomega](#) - A matcher/assertion library for Go

Code Coverage Tools

- [Go cover](#) - A tool that provides code coverage statistics for Go programs
- [Istanbul](#) - Yet another JS code coverage tool
- [cobertura](#) - A free Java tool that calculates the percentage of code accessed by tests
- [LCOV](#) - An extension of GCOV, a GNU tool which provides information about what parts of a program are actually executed
- [kcov](#) - A code coverage tester for compiled programs
- [SimpleCov](#) - A code coverage tool for Ruby with a powerful configuration library and automatic merging of coverage across test suites

Test Supporting Tools

- Mocking Libraries
 - Jest / Vitest built-in ones
 - [mock](#) - A command-line tool for mocking HTTP responses
 - [unittest.mock](#) - A library for testing in Python that allows you to replace parts of your system under test with mock objects
 - [sinon.js](#) - A standalone and test framework agnostic JavaScript test spies, stubs and mocks

- [mockery](#) - A project that creates mock implementations of Golang interfaces
- Test Data Generators
 - [Databricks Labs Data Generator](#) - A Python library for generating synthetic data within the Databricks environment using Spark
 - [generatedata.com](#) - A powerful, feature-rich, random test data generator
 - [gofakeit](#) - A random data generator written in go
 - [Fake-rs](#) - A library for generating fake data in Rust
- Multi-Environment Test Runners
 - [nox](#) - A command-line tool that automates testing in multiple Python environments, similar to tox
 - [tox](#) - A command-line driven automated testing tool for Python

Mutation Testing

- [Mutation testing](#) - A type of software testing where certain statements of the source code are changed to check if the test suite is able to find the errors
 - [cargo-mutants](#) - A tool for Rust that helps you improve your program's quality by finding places where bugs can be inserted without causing any tests to fail
 - [Stryker Mutator](#) - A tool to test your tests with mutants
 - [PIT Mutation Testing](#) - A state of the art system, providing gold standard test coverage for Java and the JVM

Build Automation

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- 3. Technology > 3.1 Software Development > Cloud Infrastructure Utilization
- Build Automation Tools
 - [GNU Make](#) - A tool which controls the generation of executables and other non-source files of a program
 - [Remake](#) - An enhanced version of GNU Make that adds improved error reporting, better tracing, profiling and a debugger
 - [makefile-graph](#) - A Go module and CLI application, which parses GNU Make's internal database and generates a graph

- [Bazel](#) - A fast, scalable, multi-language and extensible build system
- [Gradle](#) - An open-source build automation tool that is designed to be flexible enough to build almost any type of software
- [Maven](#) - A software project management and comprehension tool
- [Task](#) - A task runner / build tool that aims to be simpler and easier to use than GNU Make
- [CMake](#) - An open-source, cross-platform family of tools designed to build, test and package software
 - [CPack](#) - A tool to configure generators for binary installers and source packages
- [Meson](#) - An open source build system meant to be both extremely fast, and, even more importantly, as user friendly as possible
- [Dune](#) - A composable build system for OCaml, OCaml-related languages, and Coq
- [Rake](#) - A Make-like program implemented in Ruby
- [fpm](#) - A tool which lets you easily create packages for Debian, Ubuntu, Fedora, CentOS, RHEL, Arch Linux, and more
- Tutorials
 - [Makefile Tutorial by Example](#) - A tutorial that teaches you the basics of Makefiles

Monorepo Management

- Monorepo Tools
 - [Turborepo](#) - A high-performance build system for JavaScript and TypeScript monorepos
 - [Nx](#) - A smart, fast and extensible build system with first-class monorepo support and powerful integrations
 - [Lerna](#) - The original monorepo tool for JavaScript/TypeScript
- Resources
 - [Monorepo Tools](#) - A website with tools and resources for monorepos

Program Documentation

- Program Documentation Tools
 - [apiDoc](#) - A tool that creates a documentation from API descriptions in your source code

- [JSDoc](#) - An API documentation generator for JavaScript
- [perldoc](#) - A tool that looks up a piece of documentation in .pod format that is embedded in the perl installation tree
 - [Pod](#) - A simple-to-use markup language used for writing documentation for Perl, Perl programs, and Perl modules
- [pydoc](#) - A tool that automatically generates documentation from Python modules
 - [Docstring](#) - A string literal that appears as the first statement in a module, function, class, or method definition
- [godoc](#) - A tool that extracts and generates documentation for Go programs
- [rustdoc](#) - A tool that generates documentation for Rust projects
- [RDoc](#) - A tool that produces HTML and command-line documentation for Ruby projects
- [Javadoc](#) - A tool from Oracle for generating API documentation in HTML format from doc comments in source code

Package Dependency Management

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- JavaScript & Web Ecosystem
 - [npm CLI](#) - The world's largest software registry
 - [npm-check-updates](#) - A command-line tool that allows you to upgrade your package.json dependencies to the latest versions
 - [npmgraph](#) - A tool for exploring the npm dependency graph
 - [yarn](#) - A package manager that doubles down as project manager
 - [pNPm](#) - A fast, disk space efficient package manager
 - [Corepack](#) - A zero-runtime-dependency Node.js script that acts as a bridge between Node.js projects and the package managers they are intended to be used with during development
 - [dpmland](#) - A simple, modern and easy way to manage the Deno modules and dependencies
 - [Bun package manager](#) - A fast, npm-compatible package manager built into Bun
 - [orogene](#) - A next-generation package manager for the JavaScript ecosystem

- [vlt](#) - A JavaScript package registry for teams that move fast, serving npm-compatible packages together with a secure npm mirror and a fully resolved dependency graph
- Python Development
 - [pip](#) - The package installer for Python
 - [poetry](#) - A tool for dependency management and packaging in Python
 - [pdm](#) - A modern Python package and dependency manager supporting the latest PEP standards
 - [uv](#) - An extremely fast Python package and project manager, written in Rust
- Systems & Native Languages
 - [go mod](#) - A tool for managing Go source code
 - [Cargo](#) - The Rust package manager
 - [Conan](#) - A dependency and package manager for C and C++ languages
 - [vcpkg](#) - A free C/C++ package manager for acquiring and managing libraries
 - [bpkg](#) - A lightweight bash package manager
- JVM & .NET Frameworks
 - [Maven](#) - A software project management and comprehension tool
 - [Gradle](#) - An open-source build automation tool that is designed to be flexible enough to build almost any type of software
 - [NuGet CLI](#) - The package manager for .NET
- Other Language Ecosystems
 - [RubyGems CLI](#) - The official package manager for Ruby
 - [Bundler](#) - A tool that provides a consistent environment for Ruby projects
 - [LuaRocks CLI](#) - The package manager for Lua modules
 - [cpanminus](#) - A tool to get, unpack, build and install modules from CPAN
 - [stack](#) - A cross-platform program for developing Haskell projects
 - [opam](#) - A source-based package manager for the OCaml community
 - [Pub](#) - The official package manager for Dart and Flutter
 - [Hex](#) - The package manager for the Erlang ecosystem

Virtual Environment

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science

- Virtual Environment Managers

- [Python venv](#) - A module for the creation of virtual environments
- [pyenv](#) - A tool for simple Python version management
- [nodeenv](#) - A tool to create isolated node.js environments
- [nvm](#) - A POSIX-compliant bash script to manage multiple active node.js versions
- [nvm-windows](#) - A node.js version manager for Windows
- [rv](#) - A simple and powerful Ruby version manager written in Rust
- [frum](#) - A fast and modern Ruby version manager written in Rust
- [g](#) - A simple Go version manager, gluten-free
- [perlbrew](#) - A tool to manage multiple perl installations in your \$HOME directory
- [asdf](#) - A tool version manager
- [mise](#) - A polyglot tool version manager
- [tenv](#) - A versatile version manager for OpenTofu, Terraform, Terragrunt and Atmos

11 - Specialized Development Domains

Business & Productivity Application SDKs

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Service Utilization

Project & Work Management

- [Python Jira](#) - A Pythonic interface to the JIRA REST APIs
- [Notion SDK for JavaScript](#) - An official client for the Notion API
- [Linear SDK](#) - A typed TypeScript SDK for interacting with Linear's GraphQL API for planning and building products
- [Asana Python SDK](#) - The official Python client library for the Asana API that provides programmatic access to Asana's project management platform
- [py-trello](#) - A Python API wrapper that represents Trello objects (boards, lists, cards) as corresponding Python objects with cached attributes

Collaboration & Communication

- [Python Slack SDK](#) - A software development kit that helps Python developers build apps that integrate with Slack
- [Slack API in Go](#) - A Go library for the Slack API
- [discord.js](#) - A powerful Node.js module that allows you to interact with the Discord API very easily
- [Microsoft Teams JavaScript client library](#) - A library that helps you create hosted experiences in Teams, Microsoft 365 app, and Outlook, where your app content is hosted in an iFrame
- [Notify](#) - A dead simple Go library for sending notifications to various messaging services
- [Twilio Node.js SDK](#) - The official Node.js helper library for the Twilio API, providing developers with tools to integrate Twilio's communication services
- [Zoom Meeting SDK](#) - A WebAssembly-based SDK that embeds Zoom Meeting and Webinar experiences in web applications with support for Angular, React, and Vue.js

Cloud Storage & File Management

- [Dropbox JavaScript SDK](#) - The official Dropbox SDK for JavaScript that provides API V2 access for Node.js and browser applications with OAuth support and TypeScript compatibility
- [Box Node SDK](#) - A JavaScript interface for interacting with the Box API that provides complete coverage of Box's API ecosystem with features including authentication methods and automatic retries
- [Airtable.js](#) - The official JavaScript library that provides a simple way to access and interact with your Airtable data through its RESTful API

Email & Marketing Automation

- [SendGrid Node.js SDK](#) - The official Twilio SendGrid Node.js API library for quickly and easily using the SendGrid Web API v3
- [Mailchimp Marketing Node.js SDK](#) - The official Node.js client library for the Mailchimp Marketing API that supports authentication via Basic Auth or OAuth2

Payments & Finance

- [Stripe Node.js SDK](#) - A library that provides convenient access to the Stripe API from

server-side JavaScript applications with TypeScript support and automatic retries

CRM & Customer Support

- [JSforce](#) - A Salesforce API library for JavaScript applications
- [HubSpot Node.js SDK](#) - An official SDK that provides access to HubSpot's V3 API for managing CRM objects, files, OAuth, and other platform features
- [node-zendesk](#) - A trusted API client library for Node.js and the browser that provides seamless integration with Zendesk's Customer Support Platform

Enterprise Workspaces

- Microsoft 365 Development
 - [PnPjs](#) - A collection of fluent libraries for consuming SharePoint, Graph, and Office 365 REST APIs
 - [SharePoint Framework \(SPFx\)](#) - A page and web part model that provides full support for client-side SharePoint development, easy integration with SharePoint data, and extending Microsoft Teams and Microsoft Viva
 - [CLI for Microsoft 365](#) - A cross-platform command line interface that enables you to manage your Microsoft 365 tenant and SharePoint Framework projects on any platform
 - [Microsoft Graph](#) - The gateway to data and intelligence in Microsoft 365
 - [Work-IQ](#) - An MCP (Model Context Protocol) server and CLI for accessing Microsoft 365 data
- Google Workspace Development
 - [Google Workspace CLI](#) - A command-line tool for Drive, Gmail, Calendar, Sheets, Docs, Chat, Admin, and more, dynamically built from Google Discovery Service and including AI agent skills
 - [Google APIs Node.js Client](#) - A Node.js client library for using Google APIs with support for OAuth 2.0, API Keys, and JWT token authentication

Developer Tools Integration SDKs

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development

Version Control & DevOps

- [Octokit.js](#) - The all-batteries-included GitHub SDK for Browsers, Node.js, and Deno
- [PyGithub](#) - A Python library that enables you to manage GitHub resources such as repositories, user profiles, and organizations through the GitHub REST API
- [go-github](#) - A Go client library for accessing the GitHub v3 REST API, providing comprehensive coverage of GitHub's endpoints
- [python-gitlab](#) - A wrapper for the GitLab API written in Python
- [GitLab Go Client](#) - An official Go library for interacting with the GitLab API, providing comprehensive support for GitLab resources
- [GitLab Ruby Gem](#) - A Ruby wrapper and CLI for the GitLab REST API that allows developers to interact with GitLab programmatically
- [Atlassian Python API](#) - A Python library that provides a simple wrapper for interacting with Atlassian product REST APIs including Jira, Confluence, and Bitbucket

Container & Orchestration

- [Docker SDK for Python](#) - A Python library for the Docker Engine API that lets you do anything the docker command does from within Python applications
- [Docker SDK for Go](#) - The official Go client for the Docker Engine API
- [Kubernetes client-go](#) - The official Go client library for interacting with Kubernetes clusters, providing clientsets, dynamic clients, and controller-building tools
- [Kubernetes Python Client](#) - The official Python library for interacting with the Kubernetes API, enabling programmatic cluster management and resource operations

CI/CD & Automation

- [python-jenkins](#) - A Python wrapper for the Jenkins REST API that provides a conventionally pythonic way to control and automate Jenkins servers
- [GitHub Actions Toolkit](#) - A set of packages that provides functions and utilities to make creating GitHub Actions easier, including features for inputs/outputs and file operations
- [Terraform Plugin SDK](#) - A framework that enables building Terraform plugins (providers) to manage service providers and custom in-house solutions

Computer Graphics & Game Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Front-end System Development

3D Graphics

- [Three.js](#) - A JavaScript 3D library that makes it easy to create and display animated 3D computer graphics in a web browser using WebGL
- [GSAP](#) - A robust JavaScript toolset that turns developers into animation superheroes

2D Graphics

- [PixiJS](#) - An advanced, open-source 2D rendering engine designed for creating stunning visual experiences on the web

Graphics APIs

- [WebGL](#) - A cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL ES
- [OpenGL](#) - A cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics
- [Vulkan](#) - A low-overhead, cross-platform, 3D graphics and computing API
 - [nvk](#) - Vulkan headers for the Go programming language

Binary & Media Processing

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Back-end System Development

Binary Format Tools

- [file \(command\)](#) - A file type guesser
- [hexdump](#) - A filter which displays the specified files, or the standard input, in a user specified format
- [xxd](#) - The well-known hex-dump-type utility
- [bed](#) - A binary editor written in Go
- [fq](#) - A tool inspired by jq that lets you poke at binary formats

- [ELF format](#) - A common standard file format for executable files, object code, shared libraries, and core dumps
- [BinData](#) - A declarative way to read and write structured binary data in Ruby

Data Serialization

- [Protobuf](#) - A language-neutral, platform-neutral, extensible mechanism for serializing structured data
- [MessagePack](#) - An efficient binary serialization format

Image & Media Formats

- [JPEG](#) - A commonly used method of lossy compression for digital images
- [PNG](#) - A raster-graphics file format that supports lossless data compression
- [Webp](#) - A raster graphics file format developed by Google intended as a replacement for JPEG, PNG, and GIF file formats
- [MPEG-4](#) - A method of defining compression of audio and visual digital data
- [High Efficiency Video Coding](#) - A video compression standard, designed as a successor to the widely used Advanced Video Coding (AVC)

Image & Media Processing

- Tools
 - [Swatchify](#) - A tool that uses k-means clustering to extract dominant colors from images
 - [exiftool](#) - A command-line application and Perl library for reading and writing meta information in files
 - [ImageMagick](#) - A free, open-source software suite, used for editing and manipulating digital images
 - [FFmpeg](#) - A complete, cross-platform solution to record, convert and stream audio and video
- Libraries
 - [go-mp4](#) - A Go library that provides low-level I/O interfaces for MP4
 - [Native WebP for Go](#) - A native WebP encoder written entirely in Go, with no dependencies on libwebp or other external libraries
 - [Pillow](#) - The friendly PIL (Python Imaging Library) fork that adds image processing capabilities to your Python interpreter

- [RMagick](#) - A binding from Ruby to the ImageMagick image manipulation library
- [pure_jpeg](#) - A pure Ruby JPEG encoder and decoder library with no native dependencies

Compression & Archiving

- Tools
 - [GNU Gzip](#) - A popular data compression program
 - [GNU tar](#) - A program that provides the ability to create tar archives, as well as various other kinds of manipulation
 - [Info-Zip](#) - A set of open-source software for handling ZIP archives
 - [P7ZIP](#) - A port of 7za.exe for POSIX systems
- Libraries
 - [Python Data Compression and Archiving libs](#) - The modules that support data compression and the creation and reading of archives
 - [Go compress libs](#) - A package that defines a common interface to compression and decompression algorithms
 - [Go archive libs](#) - A package that defines a common interface for accessing archived file formats
 - [JSZip](#) - A javascript library for creating, reading and editing .zip files
 - [Ruby module Zlib](#) - A module that contains classes for compressing and decompressing streams, and for working with gzip-format files
 - [zlib](#) - A free, general-purpose, legally unencumbered lossless data-compression library for use on virtually any computer hardware and operating system
 - [zlib-rs](#) - A safer zlib
 - [snappy](#) - A compression/decompression library that aims for very high speeds and reasonable compression
 - [rubyzip](#) - A ruby library for reading and writing zip files
 - [klauspost/compress](#) - A package that provides various compression algorithms including zstandard, S2, optimized deflate (gzip, zip, zlib), snappy, and entropy encoding for Go

Document Processing

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Back-end System Development

Business Document Formats

- [PDF](#) - A file format developed by Adobe to present documents in a manner independent of application software, hardware, and operating systems
- [PDF/A](#) - An ISO-standardized version of the Portable Document Format (PDF) specialized for use in the archiving and long-term preservation of electronic documents
- [Office Open XML](#) - A zip-based XML-based file format for representing spreadsheets, charts, presentations and word processing documents
- [OpenDocument](#) - A zip-compressed, XML-based file format for spreadsheets, charts, presentations and word-processing documents

PDF Processing

- Tools
 - [Docling](#) - A powerful library which simplifies document processing, parsing diverse formats
 - [Ghostscript](#) - A suite of software based on an interpreter for Adobe Systems' PostScript and Portable Document Format page description languages
 - [qpdf](#) - A command-line tool and C++ library that performs content-preserving transformations on PDF files
 - [pdftk server](#) - A command-line tool for working with PDFs
 - [pdfcpu](#) - A PDF processor written in Go
 - [MinerU](#) - A high-quality tool for convert PDF to Markdown and JSON
 - [Nano-PDF](#) - A command-line tool that enables users to edit PDF slides using natural language instructions, powered by AI vision models, with multi-page parallel editing and non-destructive text-layer preservation through OCR
- Libraries
 - [Folio](#) - A modern PDF library for Go that includes a layout engine, HTML-to-PDF conversion, support for forms, digital signatures, barcodes, and PDF/A compliance
 - [Poppler](#) - A PDF rendering library based on the xpdf-3.0 code base
 - [PDF.js](#) - A general-purpose, web standards-based platform for parsing and rendering PDFs
 - [pypdf](#) - A pure-python PDF library capable of splitting, merging, cropping, and

transforming the pages of PDF files

- [PyMuPDF](#) - A high-performance Python library for data extraction, analysis, conversion & manipulation of PDF (and other) documents
- [pdfplumber](#) - A Python library to plumb a PDF for detailed information about each text character, rectangle, and line, plus table extraction and visual debugging
- [Prawn PDF](#) - A fast, nimble PDF generator for Ruby
- [ReportLab](#) - The Open Source Python library for generating PDFs and graphics

Office Document Processing

- Tools
 - [libreoffice cli](#) - The command-line interface for the LibreOffice office suite
 - [markitdown](#) - A lightweight Python utility for converting various files to Markdown for use with LLMs and related text analysis pipelines
 - [xlsx2csv](#) - A fast and easy way to convert XLSX files to CSV
 - [docx2txt](#) - A pure python-based command line tool to extract text from docx files
 - [pptx2md](#) - A simple tool for converting pptx to markdown
- Libraries
 - [python-pptx](#) - A Python library for creating and updating PowerPoint (.pptx) files
 - [PptxGenJS](#) - A JavaScript library for building PowerPoint presentations that works with Node.js, React, and web browsers
 - [Excelize](#) - A Go library for reading and writing XLSX/XLSM/XLTM files
 - [Roo](#) - A library that can access the contents of various spreadsheet files

CLI/TUI Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Computer Science
- 3. Technology > 3.1 Software Development > Team Development
- Bash
 - [built-in getopts etc.](#) - A set of commands that are part of the shell itself
 - [tput](#) - A command to initialize a terminal or query the terminfo database
 - [dialog](#) - A program that can be used to create nice user interfaces for shell

scripts

- [Gum](#) - A tool for glamorous shell scripts
- [FIGlet](#) - A program for making large letters out of ordinary text
- [lolcat](#) - A program that concatenates files, or standard input, to standard output and adds rainbow coloring
- [cfonts](#) - A tool to print sexy fonts in your console

• Perl

- [Getopt::Long](#) - A module that implements an extended getopt function called GetOptions()
- [Term::ANSIColor](#) - A module to colorize text using ANSI escape sequences
- [Text::ANSITable](#) - A module to create a formatted table using ASCII characters and ANSI colors

• Python

- [argparse](#) - The module for parsing command-line arguments
- [getopt](#) - The C-style parser for command line options
- [click](#) - A Python package for creating beautiful command line interfaces in a composable way with as little code as necessary
- [Colorama](#) - A simple cross-platform API for printing colored terminal text from Python
- [Typer](#) - A library for building CLI applications that users will love using and developers will love creating
- [Asciimatics](#) - A package that provides a cross-platform, full-screen terminal API for building text-based user interfaces
- [Python Prompt Toolkit](#) - A library for building powerful interactive command line and terminal applications in Python
- [Questionary](#) - A Python library for building interactive command line prompts
- [Urwid](#) - A console user interface library for Python
- [Textual](#) - A Rapid Application Development framework for Python, built by Textualize.io
 - [Rich](#) - A Python library for rich text and beautiful formatting in the terminal

• Ruby

- [OptionParser](#) - A class for command-line option analysis
- [colorize](#) - A gem for colorizing text using ANSI escape sequences

- [TTY](#) - A suite of gems that provide a wide range of tools for building interactive command-line applications
- [thor](#) - A toolkit for building powerful command-line interfaces
- [dry-cli](#) - A general-purpose framework for developing Command Line Interface (CLI) applications that represents commands as objects and offers support for arguments, options, and forwarding variadic arguments to sub-commands
- [Clamp](#) - A minimal framework for command-line utilities that handles parsing command-line arguments and generating help
- [TableTennis](#) - A library for printing stylish tables in your terminal, with auto-theming, layout fitting, and numeric column coloring
- Javascript
 - [yargs](#) - A library for building interactive command line tools by parsing arguments and generating an elegant user interface
 - [minimist](#) - A tool to parse argument options
 - [chalk](#) - A terminal string styling tool
 - [cli-progress](#) - An easy to use progress-bar for command-line/terminal applications
 - [FIGlet.js](#) - A FIG Driver written in JavaScript which aims to fully implement the FIGfont spec
 - [Ink](#) - A React-based library for building CLI applications
 - [gradient-string](#) - A library for creating beautiful gradients in terminal output
- Go
 - [Fang](#) - The CLI starter kit. A small, experimental library for batteries-included Cobra applications
 - [Bubble Tea](#) - A Go framework for building terminal apps based on The Elm Architecture
 - [Lip Gloss](#) - A declarative library for defining styles and layouts in terminal applications
 - [Bubbles](#) - A collection of common terminal user interface components
 - [Huh](#) - A simple, powerful, and elegant TUI library for building terminal forms and prompts
 - [pflag](#) - A drop-in replacement for Go's flag package, implementing POSIX/GNU-style --flags
 - [color](#) - A package for Go that lets you use colorized outputs in terms of ANSI escape sequences

- [Cobra](#) - A framework for creating powerful modern CLI applications
- [urfave/cli](#) - A simple, fast, and fun package for building command line apps in Go
- [viper](#) - A complete configuration solution for Go applications
- [Wish](#) - A tiny SSH server for your programs
- [Wishlist](#) - An SSH directory for your private hostkeys and favorite SSH commands
- [go-tui](#) - A framework for building declarative terminal user interfaces (TUIs) in Go
- [asciigraph](#) - A Go package to make lightweight ASCII line graphs in command line apps with no other dependencies
- Rust
 - [clap](#) - A full featured, fast Command Line Argument Parser for Rust
 - [Ratatui](#) - A Rust library for cooking up delicious terminal user interfaces
 - [R3BL](#) - A suite of libraries for building modern terminal apps with Rust
 - [Ansic](#) - A modern, efficient and compile time ansi macro and utilities crate for Rust
- C
 - [ncurses](#) - A programming library providing an application programming interface (API) that allows the programmer to write text-based user interfaces in a terminal-independent manner

Desktop App Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Front-end System Development

Client OS & Environments

- Windows Environment
 - Package Management & Administration
 - [Chocolatey](#) - The package manager for Windows
 - [Scoop](#) - A command-line installer for Windows
 - [WinGet](#) - A command line tool enabling users to discover, install, upgrade, remove and configure applications on Windows

- [gsudo](#) - A Sudo for Windows, with a similar user-experience to the original Sudo
- Productivity & Automation
 - [AutoHotKey](#) - A free, open-source scripting language for Windows that allows users to easily create small to complex scripts
 - [Clavier+](#) - Allows to trigger actions with keyboard shortcuts
 - [WinSSHTerm](#) - A tabbed SSH solution for Windows, combining PuTTY, WinSCP and VcXsrv
- Android on Linux
 - [Waydroid](#) - A container-based approach to boot a full Android system on regular GNU/Linux systems running Wayland based desktop environments

GUI Systems & Windowing

- Display Servers
 - [X.org](#) - An open source implementation of the X Window System
 - [Wayland](#) - A replacement for the X11 window system protocol and architecture
- Desktop Environments
 - [GNOME](#) - An easy and elegant way to use your computer, designed to put you in control and get things done
 - [Xfce](#) - A lightweight desktop environment for UNIX-like operating systems
- Window Managers
 - [openbox](#) - A highly configurable, next generation window manager

Desktop GUI Toolkits

- Standard Toolkits
 - [Tk](#) - A graphical user interface toolkit
 - [tkinter](#) - The standard Python interface to the Tcl/Tk GUI toolkit
 - [CustomTkInter](#) - A python UI-library based on Tkinter, which provides new, modern and fully customizable widgets
 - [GTK](#) - A free and open-source cross-platform widget toolkit for creating graphical user interfaces
 - [pygobject](#) - A set of Python bindings for the GLib, GObject, GIO and GTK object-oriented C libraries

- **Compiled & Canvas-Based (Custom Rendering)**
 - [Slint](#) - A declarative GUI toolkit to build native user interfaces for Rust, C++, or JavaScript apps
 - [Gio](#) - A library for writing cross-platform immediate mode GUI-s in Go
 - [Fyne](#) - An easy to learn toolkit for creating graphical apps for desktop, mobile and web
- **Web-Technology Based**
 - **Chromium Bundling**
 - [Electron](#) - A framework for building desktop applications using JavaScript, HTML, and CSS
 - **System WebView (Hybrid)**
 - [Tauri](#) - A toolkit that helps developers make applications for the major desktop platforms
 - [Wails](#) - A tool that enables developers to build desktop applications using Go and web technologies
 - [Microsoft Edge WebView2](#) - A control that allows you to embed web technologies (HTML, CSS, and JavaScript) in your native apps by using Microsoft Edge as the rendering engine

Desktop App Testing & Automation

- [xa11y](#) - A Playwright-style library for driving native desktop apps on macOS, Windows, and Linux

Installation & Packaging

- [NSIS](#) - A professional open source system to create Windows installers
- [Windows App CLI \(winapp\)](#) - A command-line interface for managing Windows SDKs, packaging apps with MSIX, generating app identity, manifests, and certificates for cross-platform frameworks
- [PyInstaller](#) - A tool that bundles a Python application and all its dependencies into a single package

Mobile App Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Front-end System Development

Mobile Platforms & Native SDKs

- [Android](#) - A mobile operating system based on a modified version of the Linux kernel and other open-source software
 - [Android Studio](#) - The official integrated development environment for Google's Android operating system
- [iOS](#) - A mobile operating system created and developed by Apple Inc. exclusively for its hardware

Mobile Cross-Platform Frameworks

- Canvas-Based (Custom Rendering)
 - [Flutter](#) - An open-source UI software development kit created by Google
- Native Widget Bridge
 - [React Native](#) - A framework for building native apps with React
 - [Expo](#) - A framework and a platform for universal React applications that run natively on Android, iOS, and the web
 - [.NET MAUI](#) - A cross-platform framework for creating native mobile and desktop apps with C# and XAML
- Web-Technology Based
 - [Capacitor](#) - A cross-platform native runtime that makes it easy to build performant mobile apps, desktop apps, and Progressive Web Apps
 - [Ionic Framework](#) - An open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies

Mobile DevOps & Testing

- [fastlane](#) - The easiest way to automate beta deployments and releases for your iOS and Android apps
- [Appium](#) - An open source test automation framework for use with native, hybrid and mobile web apps

Application Services & Features

- Notifications
 - [Firebase Cloud Messaging](#) - A cross-platform messaging solution that lets you reliably send messages at no cost
 - [Apple Push Notification service](#) - The service for propagating push notifications

from a developer's server to their apps on Apple devices

- Device Hardware/OS Integration
 - [GPS](#) - A satellite-based radionavigation system owned by the United States government and operated by the United States Space Force
 - [QR code](#) - A type of matrix barcode invented in 1994 by the Japanese company Denso Wave
 - [libqrencode](#) - A fast and compact QR Code encoding library
 - [Pure python QR Code generator](#) - A library that provides a simple way to create QR codes in Python
 - [QR code payment](#) - A contactless payment method where a payment is performed by scanning a QR code from a mobile app
- Backend-as-a-Service (BaaS)
 - [Firebase](#) - An app development platform that helps you build and grow apps and games users love
 - [Supabase](#) - A Postgres development platform that provides a database, authentication, instant APIs, edge functions, real-time subscriptions, storage, and vector embeddings
 - [AWS Amplify](#) - A complete solution that lets frontend web and mobile developers easily build, ship, and host full-stack applications on AWS

Internet of Things (IoT)

Relevant DSS-P Skills

- 3. Technology > 3.2 Digital Technology > Physical Computing

IoT Concepts

- [Internet of things \(IoT\)](#) - The network of physical objects—'things'—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet
- [Edge computing](#) - A distributed computing paradigm that brings computation and data storage closer to the sources of data
- [Machine to machine](#) - The direct communication between devices using any communications channel, including wired and wireless
- [Cyber-physical system](#) - A mechanism controlled or monitored by computer-based algorithms, tightly integrated with the internet and its users

- [Digital twin](#) - A digital model of an intended or actual real-world physical product, system, or process that serves as its digital counterpart for purposes such as simulation, testing, monitoring, and maintenance
- [Firmware](#) - A specific class of computer software that provides the low-level control for a device's specific hardware
- [Over-the-air update](#) - The wireless delivery of new software, firmware, or other data to mobile devices

Communication Standards

- [Wi-Fi](#) - A family of wireless network protocols, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access
- [Bluetooth Low Energy](#) - A wireless personal area network technology designed to provide considerably reduced power consumption and cost while maintaining a similar communication range to classic Bluetooth
- [Zigbee](#) - An IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios
- [Near-field communication](#) - A set of communication protocols that enables communication between two electronic devices over a distance of 4 cm (1+1/2 in) or less

IoT Hardware Platforms

- [Raspberry Pi](#) - A small, affordable computer for you to use and learn with
- [Arduino](#) - An easy-to-use, open-source hardware and software platform that enables innovation and creativity in electronics projects
- [ESP32](#) - A feature-rich microcontroller unit with integrated Wi-Fi and Bluetooth connectivity, designed for IoT applications with ultra-low power consumption
- [ESP8266](#) - A cost-effective and highly integrated Wi-Fi MCU for IoT applications featuring a 32-bit Tensilica processor with low power consumption
- [BeagleBone](#) - A low-cost, community-supported development platform featuring a 1GHz ARM Cortex-A8 processor that boots Linux in under 10 seconds
- [STM32](#) - A family of 32-bit microcontrollers based on Arm Cortex-M processors designed to offer high performance, real-time capabilities, and low-power operation
- [Nordic nRF52](#) - A family of Bluetooth Low Energy System-on-Chips built around 64

MHz ARM Cortex-M4 processors that support multiprotocol wireless connectivity

- [Particle](#) - An integrated IoT platform-as-a-service that provides edge-to-cloud infrastructure for developing, connecting, and managing intelligent devices
- [BBC micro:bit](#) - A pocket-sized programmable computer designed to inspire children to create their best digital future through hands-on coding and creative technology projects
- [Teensy](#) - A compact USB-based microcontroller development board compatible with Arduino software and capable of implementing many project types
- [Adafruit Feather](#) - A family of compact, portable microcontroller boards designed as a standard for embedded projects with built-in battery connectors and modular expansion capabilities

IoT Cloud Platforms

- [Azure IoT Hub](#) - A managed cloud-based service that serves as a central message hub for communication between an IoT application and its connected devices
- [AWS IoT Core](#) - A managed cloud service that enables you to easily and securely connect devices to the cloud and manage device fleets at scale
- [AWS IoT Greengrass](#) - An open-source edge runtime and cloud service for building, deploying, and managing device software
- [ThingWorx](#) - A comprehensive industrial IoT application development platform designed to help organizations connect devices, analyze data, and deliver real-time insights through customizable applications
- [Balena](#) - A container-based IoT device management platform that enables developers to build, deploy, manage, and scale fleets of Linux devices with secure OTA updates and remote access
- [Arduino Cloud](#) - An all-in-one IoT platform that enables users to build, control, and monitor connected projects from anywhere with dashboards and remote access
- [ThingSpeak](#) - An IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud with MATLAB integration
- [Losant](#) - An enterprise IoT platform that enables organizations to build scalable connected solutions by collecting, integrating, and visualizing data from devices and systems in real-time
- [Adafruit IO](#) - The easiest way to get your projects onto the Internet of Things, serving as a platform for web-based microcontroller interaction and datalogging
- [ThingsBoard](#) - An open-source IoT platform for data collection, processing, visualization, and device management that enables device connectivity via industry

standard protocols

- [Ubidots](#) - A cloud-based Industrial IoT platform that enables businesses to connect, monitor, visualize, and act on sensor data in real-time through customizable dashboards and automated workflows

Low-Code & No-Code Development

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Service Utilization

Business Application Platforms

- [Microsoft Power Apps](#) - A platform to rapidly and efficiently build professional-grade apps for any device
- [AppSheet](#) - The no-code platform that helps build powerful applications and automations to boost productivity
- [OutSystems](#) - A high-performance low-code platform that accelerates the development of critical enterprise applications

Workflow & Integration Automation

- [n8n](#) - A fair-code licensed workflow automation tool that combines AI capabilities with business process automation
- [Microsoft Power Automate](#) - An end-to-end automation solution built for enterprise to optimize business processes
- [Zapier](#) - A tool that allows you to connect your apps and automate workflows without any coding knowledge
- [Microsoft Copilot Studio](#) - A graphical low-code tool for creating and maintaining copilots using generative AI and a library of prebuilt connectors and actions

Web Content & Portal Builders

- [Microsoft Power Pages](#) - A platform to quickly create enterprise-grade AI-driven business portals with built-in agents
- [Webflow](#) - A browser-based design tool that gives you the power to build production-ready websites without coding

12 - Personal Skills

Foundational Thinking & Logic

Relevant DSS-P Skills

- 5. Personal Skill > 5.2 Conceptual Skill > Creative Problem Solving
- 5. Personal Skill > 5.2 Conceptual Skill > Critical Thinking

Logic

- **Logic** - The study of correct reasoning
 - **Logical reasoning** - A mental activity that aims to arrive at a conclusion in a rigorous way
 - Deductive, Inductive, Abductive and Analogical
 - **First principle** - A basic proposition or assumption that cannot be deduced from any other proposition or assumption
- Branches of Logic
 - **Informal Logic** - A broad term for any of the various methods of analyzing and evaluating arguments used in everyday life

Informal Logic

- **Argument** - The central object of study in informal logic; a series of statements (premises) intended to determine the degree of truth of another statement (the conclusion)
 - **Enthymeme** - An argument in which one premise is not explicitly stated, a common feature of real-world reasoning
- Criteria for Argument Evaluation
 - **Fallacy** - The use of invalid or otherwise faulty reasoning in the construction of an argument that may appear to be well-reasoned if unnoticed
 - **Category mistake** - An error in reasoning where a property is ascribed to a thing that could not possibly have that property
- Related Fields
 - **Rhetoric** - The art of persuasion
 - **Critical Thinking** - The process of analyzing available facts, evidence, observations, and arguments to make sound conclusions or informed choices

Mathematical Logic

Relevant DSS-P Skills

- 5. Personal Skill > 5.2 Conceptual Skill > Critical Thinking

Foundational Concepts

- **Formal system** - An abstract structure and formalization of an axiomatic system used for deducing, using rules of inference, theorems from axioms by a set of inference rules
- **Gödel's incompleteness theorems** - The two theorems of mathematical logic that demonstrate the inherent limitations of every formal axiomatic system capable of modelling basic arithmetic
- Logic Principles
 - **De Morgan's laws** - A pair of transformation rules that are both valid rules of inference
 - **Law of noncontradiction** - The law that states that for any given proposition, the proposition and its negation cannot both be simultaneously true
 - **Law of excluded middle** - The principle that for every proposition, either this proposition or its negation is true
 - **Peirce's law** - The principle in classical logic that the law of excluded middle holds for any proposition
 - **Proof by contradiction** - A form of indirect proof that establishes the truth of a proposition by showing that assuming the proposition to be false leads to a contradiction

Logical Systems

- **Propositional calculus** - A branch of logic that deals with propositions (which can be true or false) and relations between propositions, including the construction of arguments based on them
 - conjunction, disjunction, implication, biconditional and negation
 - **Tautology** - A formula that is true regardless of the interpretation of its component terms, with only the logical constants having a fixed meaning
- **First order logic** - A collection of formal systems used in mathematics, philosophy, linguistics, and computer science
 - universal quantification and existential quantification

- [Higher order logic](#) - A form of logic that is distinguished from first-order logic by additional quantifiers and, sometimes, stronger semantics
- [Modal logic](#) - A type of logic that is used to represent statements about possibility and necessity

Branches of Mathematical Logic

- [Set theory](#) - The branch of mathematical logic that studies sets, which can be informally described as collections of objects
- Naive set theory
 - [Set](#) - A collection of different things; these things are called elements or members of the set and are typically mathematical objects of any kind
 - [Function \(a.k.a. Map\)](#) - A binary relation between two sets that associates every element of the first set to exactly one element of the second set
 - [Operation](#) - A function from a set to itself
 - [Idempotence](#) - The property of certain operations in which they can be applied multiple times without changing the result beyond the initial application
 - [Partition of a set](#) - A grouping of a set's elements into non-empty, disjoint subsets (called "blocks" or "cells") such that every element is in exactly one subset
 - [Equivalence relation](#) - A binary relation (reflexive, symmetric, and transitive) that partitions a set into disjoint equivalence classes
- Axiomatic set theory
 - [Zermelo–Fraenkel set theory](#) - An axiomatic system that was proposed in the early twentieth century in order to formulate a theory of sets free of paradoxes such as Russell's paradox
 - Ordinals & Cardinals
- [Type Theory](#) - A formal system that provides an alternative foundation for mathematics (like Set Theory) and is the basis for typed functional programming and proof assistants
 - [Curry-Howard correspondence](#) - The direct relationship between computer programs and mathematical proofs
- [Proof Theory](#) - A major branch of mathematical logic that represents proofs as formal mathematical objects, facilitating their analysis by mathematical techniques
 - [Sequent calculus](#) - A deductive system for proving theorems

- [Natural deduction](#) - A kind of proof calculus in which logical reasoning is expressed by inference rules closely related to the "natural" way of reasoning
- [Computability Theory](#) - A branch of mathematical logic, computer science, and the theory of computation that originated in the 1930s with the study of computable functions and Turing degrees
 - [Lambda calculus](#) - A formal system in mathematical logic for expressing computation based on function abstraction and application
 - [Turing machine](#) - A mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules
- [Model Theory](#) - The study of the relationship between formal theories (collections of sentences in a formal language) and their models (structures in which the sentences are true)
- Logic Applications
 - [Constraint satisfaction problem](#) - Mathematical questions defined as a set of objects whose state must satisfy a number of constraints or limitations
 - [Satisfiability modulo theories](#) - The problem of determining whether a mathematical formula is satisfiable
 - [Automated theorem proving](#) - A subfield of automated reasoning and mathematical logic dealing with proving mathematical theorems by computer programs
 - [Formal verification](#) - the act of proving or disproving the correctness of a system with respect to a certain formal specification or property, using formal methods of mathematics
 - [Hoare logic](#) - A formal system with a set of logical rules for reasoning rigorously about the correctness of computer programs
- Formal Logic Tools
 - [Stanford Encyclopedia of Philosophy](#) - A reference work that organizes scholars in philosophy and related fields from around the world to create and maintain up-to-date content
 - [SMT-LIB](#) - A command language for interacting with SMT solvers via a textual interface
 - [MiniZinc](#) - A free and open-source constraint modeling language
 - [P](#) - A state machine based programming language for formally modeling and specifying complex distributed systems
 - [Lean](#) - An interactive theorem prover and programming language based on the

Calculus of Constructions

Documentation

Relevant DSS-P Skills

- 3. Technology > 3.1 Software Development > Team Development
- 5. Personal Skill > 5.1 Human Skill > Collaboration
- [Technical writing](#) - A type of writing where the author is writing about a particular subject that requires direction, instruction, or explanation
- [Divio Documentation System](#) - A framework that proposes that all documentation should be explicitly structured according to its purpose, into four distinct types: tutorials, how-to guides, technical reference and explanation

Architectural Documentation

- Diagramming Tools
 - [draw.io](#) - A technology stack for building diagramming applications, and the world's most widely used browser-based end-user diagramming software
- Diagramming as Code
 - [D2: Declarative Diagramming](#) - A modern diagram scripting language that turns text to diagrams
 - [Diagrams](#) - A Python package for drawing cloud system architectures in Python code
 - [PlantUML](#) - A tool that allows you to create diagrams from a simple textual description
 - [Mermaid](#) - A Javascript based diagramming and charting tool that renders Markdown-inspired text definitions to create and modify diagrams dynamically
 - [Kroki](#) - A free and open source service that converts plain text diagrams to images
 - [Graphviz](#) - An open source graph visualization software
 - [DOT language](#) - A plain text graph description language
 - [sfdp](#) - A scalable, multiscale force-directed layout engine for large undirected graphs that minimizes edge crossings and avoids node overlaps
 - [haphviz](#) - A Haskell library for representing, manipulating, and pretty-printing graphs in the DOT format

- [dita](#) - A small command-line utility that can convert diagrams drawn using ascii art into proper bitmap graphics
- Architectural Decision Records
 - [Architectural Decision Records \(ADRs\)](#) - A document that captures an important architectural decision made along with its context and consequences
 - [MADR \(Markdown Architectural Decision Records\)](#) - A lean template to capture architectural decisions in Markdown, making it as easy as possible to write down and to version the decisions
 - [adr-tools](#) - A command-line tool to help you manage your architectural decision records
 - [Log4brains](#) - An architecture knowledge base that lets developers log decisions directly from their IDE and publish them as a static website

Lightweight Markup & Writing Styles

- Lightweight Markup
 - [Markdown](#) - A lightweight markup language for creating formatted text using a plain-text editor
 - [CommonMark](#) - A rationalized version of Markdown syntax, with a spec and BSD-licensed reference implementations in C and JavaScript
 - [GFM \(GitHub Flavored Markdown\)](#) - A formal specification, based on the CommonMark Spec, that defines the syntax and semantics of GitHub's dialect of Markdown
 - [github-markdown-css](#) - The CSS that styles markdown rendered on GitHub
 - [markdownlint](#) - A Node.js style checker and lint tool for Markdown/CommonMark files
 - [Glow](#) - A terminal based markdown reader
 - [mdterm](#) - A terminal-based Markdown viewer written in Rust that renders Markdown files with syntax highlighting, styled formatting, and interactive navigation
 - [Grip](#) - A command-line server application that renders local README files before you push them to GitHub
 - [markmap](#) - A combination of Markdown and mindmap
 - [Marp](#) - The simplest Markdown presentation writer with plain Markdown
 - [Markdown all-in-one](#) - An all-in-one tool for Markdown (keyboard

shortcuts, table of contents, auto preview, and more)

- [Markdown Preview Enhanced](#) - A SUPER POWERFUL markdown extension for Visual Studio Code
- [Markdown Preview for \(Neo\)vim](#) - A markdown preview plugin for (neo)vim
- Guides
 - [Markdown Guide](#) - A free and open-source reference guide that explains how to use Markdown
- [DocUtils](#) - An open-source text processing system for processing plaintext documentation into useful formats, such as HTML, LaTeX, man-pages, open-document, or XML
 - [reStructuredText](#) - An easy-to-read, what-you-see-is-what-you-get plaintext markup syntax and parser system
- [AsciiDoc](#) - A lightweight markup language for writing notes, documentation, articles, books, ebooks, slideshows, web pages, man pages and blogs
 - [AsciiDoctor](#) - A fast, open source text processor and publishing toolchain for converting AsciiDoc content to HTML5, DocBook 5 (or 4.5) and other formats
- [Org Mode](#) - An authoring tool and a TODO lists manager for GNU Emacs
 - [nvim-orgmode](#) - An Orgmode clone for Neovim written in Lua
- [Wikitext](#) - The markup language that consists of the syntax and keywords used by the MediaWiki software to format a page
- Style Guides
 - [Microsoft Writing Style Guide](#) - A guide for writers creating a variety of content types, including apps and websites
 - [Google documentation style guide](#) - The editorial guidelines for writing clear and consistent technical documentation for an audience of software developers and other technical practitioners
 - [Red Hat documentation style guide](#) - The guide that provides style guidelines for Red Hat product and cross-product solution documentation
 - [Microsoft Terminology](#) - A collection of rules that define language and style conventions for specific languages
 - [List of English words](#) - A text file containing over 466k English words
- Prose Linters
 - [vale](#) - A linter for natural language/prose
 - [retext](#) - An extensible natural language processor

- [alex](#) - A tool that helps you find gender favoring, polarizing, race related, religion inconsiderate, or other unequal phrasing in text
- [write-good](#) - A naive linter for English prose
- [textlint](#) - The pluggable linting tool for text and markdown

Documentation Tooling

- Typesetting Systems
 - [Typst](#) - A new markup-based typesetting system that is designed to be as powerful as LaTeX while being much easier to learn and use
 - [Troff/Groff](#) - A typesetting system that reads plain text mixed with formatting commands and produces formatted output
 - [LaTeX](#) - A high-quality typesetting system; it includes features designed for the production of technical and scientific documentation
 - [TexLive](#) - A cross-platform, free software distribution for the TeX typesetting system
 - [PGF/TikZ](#) - A TeX macro package for generating graphics
 - [KaTeX](#) - The fastest math typesetting library for the web
 - [sphinxcontrib-katex](#) - A Sphinx extension which allows you to use KaTeX to render math in your Sphinx documentation
- Validation & Maintenance
 - [lychee](#) - A fast, async link checker written in Rust
- Converters
 - [Pandoc](#) - A universal document converter
 - [Eisvogel](#) - A pandoc LaTeX template to convert markdown files to PDF or LaTeX

Interpersonal & Team Leadership

Relevant DSS-P Skills

- 5. Personal Skill > 5.1 Human Skill > Leadership
- 5. Personal Skill > 5.1 Human Skill > Collaboration

Team Dynamics & Communication

- Team Dynamics

- [Team building](#) - A collective term for various types of activities used to enhance social relations and define roles within teams, often involving collaborative tasks
 - [Tuckman's stages of group development](#) - A model of group development that was first proposed by Bruce Tuckman in 1965
- [Group dynamics](#) - A system of behaviors and psychological processes occurring within or between social groups
- Research & Models
 - [Google Rework: Understand team effectiveness](#) - A research initiative identifying key dynamics like psychological safety, dependability, structure and clarity, meaning, and impact as crucial for successful group collaboration
 - [GRPI Model](#) - A foundational framework for designing and diagnosing team effectiveness, organizing the building blocks of a high-performing team into four layers: Goals, Roles, Processes, and Interpersonal Relationships
- Interpersonal Communication Techniques
 - [Storytelling](#) - The social and cultural activity of sharing stories, sometimes with improvisation, theatrics or embellishment
 - [Facilitation](#) - The act of designing and running a successful meeting or workshop
 - [Active listening](#) - The practice of preparing to listen, observing what verbal and non-verbal messages are being sent, and then providing appropriate feedback for the sake of showing attentiveness to the message being presented
 - [Negotiation](#) - A dialogue between two or more parties to resolve points of difference, gain an advantage for an individual or collective, or craft outcomes to satisfy various interests
- Corporate Principles & Values
 - [Amazon's Leadership Principles](#) - The set of core tenets Amazonians use daily to guide their discussions, decisions, and actions
 - [GitLab Values](#) - A set of guiding principles that define the company's culture and how its team members operate
- Professional Manifestos
 - [Manifesto for Software Craftsmanship](#) - A declaration emphasizing the importance of well-crafted software, continuous value addition, professional community, and productive partnerships in software development
- Feedback Models
 - [DESC feedback model](#) - A communication tool for providing constructive feedback by describing the behavior, expressing the impact, specifying the

desired change, and explaining the consequences

Organizational Behavior

- [Stakeholder management](#) - The process of identifying individuals or groups that are affected by a project or business venture, understanding their interests and concerns, and managing their expectations and influence
- [Contingency theory](#) - A theory that claims there is no single best way to organize a corporation, lead a company, or make decisions, asserting that the optimal course of action depends on internal and external situations
- [Holacracy](#) - A method of decentralized management and organizational governance, which claims to distribute authority and decision-making through a holarchy of self-organizing teams rather than being vested in a management hierarchy
- [Expectancy theory](#) - A theory that proposes an individual's behavior is motivated by the expected results of their actions, with the desirability of the outcome determining the selection of a specific behavior
- [Intrinsic motivation](#) - A type of motivation that arises from internal factors, such as enjoyment, curiosity, or a sense of fulfillment, where individuals engage in an activity for its own sake
- [Management 3.0](#) - An ever-evolving mindset and a collection of games, tools, and practices designed to help any worker manage the organization and improve work systems
- [PM Theory of Leadership](#) - A leadership theory that classifies leaders into two categories: performance-oriented and maintenance-oriented
- [Theory X and Theory Y](#) - The theories of human work motivation and management developed by Douglas McGregor
- [Two-factor theory](#) - A theory developed by psychologist Frederick Herzberg, it posits that job satisfaction and dissatisfaction are influenced by separate sets of factors, known as motivators and hygiene factors, which act independently of each other

Leadership Styles

- [Servant leadership](#) - A leadership philosophy where the primary aim of the individual in charge is to prioritize the needs of their team members, fostering their development and performance, rather than focusing on personal gain or traditional hierarchical authority
- [Shared leadership](#) - A leadership style that broadly distributes responsibility, allowing individuals within a team or organization to lead each other

- **Situational leadership** - A leadership model where effective leaders adapt their style to each situation, recognizing that no single approach is universally appropriate
- **Transformational leadership** - A leadership style in which leaders encourage, inspire, and motivate employees to innovate and create change that will help grow and shape the future success of the company

Individual Psychology & Performance

Relevant DSS-P Skills

- 5. Personal Skill > 5.2 Conceptual Skill > Adaptability

Personal Performance

- Mental Health
 - **Mindfulness** - The basic human ability to be fully present, aware of where we are and what we're doing, and not overly reactive or overwhelmed by what's going on around us
 - **Zen** - A school of Mahayana Buddhism that originated in China during the Tang dynasty
 - **Flow** - The mental state in which a person performing some activity is fully immersed in a feeling of energized focus, full involvement, and enjoyment in the process of the activity
 - **Defence mechanism** - Unconscious psychological processes that protect the self from anxiety-producing thoughts and feelings related to internal conflicts and external stressors
 - **Psychological resilience** - The ability to cope mentally and emotionally with a crisis, or to return to pre-crisis status quickly
 - **Occupational burnout** - A work-related phenomenon resulting from chronic workplace stress that has not been successfully managed
- Cognitive Performance / Decision Making
 - **Maslow's Hierarchy of Needs** - A conceptualisation of the needs (or goals) that motivate human behavior
 - **Cognitive bias** - A systematic pattern of deviation from norm or rationality in judgment
 - **Anchoring effect** - A psychological phenomenon in which an individual's judgments or decisions are influenced by a reference point or "anchor" which can be completely irrelevant

- [Zeigarnik effect](#) - The phenomenon that people remember unfinished or interrupted tasks better than completed tasks
- [Default mode network](#) - A large-scale brain network; known for being active when a person is not focused on the outside world and the brain is at wakeful rest
- [Situation awareness](#) - The understanding of an environment, its elements, and how it changes with respect to time or other factors
 - 1: Perception of the elements in the environment
 - 2: Comprehension or understanding of the situation
 - 3: Projection of future status
- [Vertical thinking](#) - A problem-solving approach characterized by being selective, analytical, and sequential, often relying on rational assessment and external data
- [Lateral thinking](#) - A manner of solving problems using an indirect and creative approach via reasoning that is not immediately obvious
- Related Philosophies
 - [Three Virtues](#) - The qualities of a great programmer: Laziness, Impatience, and Hubris
- Related Books
 - [Thinking, Fast and Slow](#) - A 2011 book by psychologist Daniel Kahneman

Social Performance

- Social Psychology
 - [Psychological safety](#) - The belief that one will not be punished or humiliated for speaking up with ideas, questions, concerns, or mistakes
 - [Trust](#) - The belief that another person will do what is expected
 - [Collective intelligence](#) - The shared or group intelligence that emerges from the collaboration, collective efforts, and competition of many individuals and appears in consensus decision making
 - [Groupthink](#) - A psychological phenomenon that occurs within a group of people in which the desire for harmony or conformity in the group results in an irrational or dysfunctional decision-making outcome
 - [Bystander effect](#) - A social psychological theory that states that individuals are less likely to offer help to a victim when there are other people present
 - [Dunbar's number](#) - A suggested cognitive limit to the number of people with

whom one can maintain stable social relationships

- **Norm of reciprocity** - The social expectation that people repay in kind what others have done for them
- Illustrative Concepts
 - **Broken windows theory** - A criminological theory that states that visible signs of crime, anti-social behavior, and civil disorder create an urban environment that encourages further crime and disorder, including serious crimes
 - **Stone soup story** - A European folk story in which hungry strangers convince the people of a town to each share a small amount of their food in order to make a meal
 - **Boiling frog apologue** - An apologue describing a frog being slowly boiled alive

Timeline - 1930-89

1930s

 Lambda calculus is a formal system in mathematical logic for expressing computation based on function abstraction and application using variable binding and substitution, introduced by mathematician Alonzo Church in the 1930s as part of an investigation into the foundations of mathematics, though the original system was shown to be logically inconsistent in 1935.

 A Turing machine is a mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules, invented by Alan Turing in 1936 and named by his doctoral advisor Alonzo Church in a later review.

1940s

 Maslow's hierarchy of needs is a theory of psychological health introduced by Abraham Maslow in his 1943 paper "A Theory of Human Motivation," published in the journal *Psychological Review*, proposing that human needs can be classified into a hierarchy ranging from basic physiological needs to self-actualization.

 In computer programming, assembly language is any low-level programming language with a very strong correspondence between the instructions in the language and the architecture's machine code instructions, with the first assembly code used to represent machine code instructions appearing in Kathleen and Andrew Donald Booth's 1947 work.

1950s

🧠 The Turing test, originally called the imitation game by Alan Turing in 1950, is a test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human.

📖 Regular expressions originated in 1951, when mathematician Stephen Cole Kleene described regular languages using his mathematical notation called regular events.

⚙️ The caesium atomic clock was first built in 1955 by Louis Essen at the National Physical Laboratory (NPL) in the UK, providing a form of timekeeping far more stable and precise than astronomical observations, and laying the foundation for the modern definition of the SI second and Coordinated Universal Time (UTC).

🎭 Parkinson's law is the adage that "work expands so as to fill the time available for its completion," first articulated by Cyril Northcote Parkinson as a humorous essay published in *The Economist* in 1955, drawing on his experience of bureaucratic growth in the British Civil Service.

📖 Lisp is a family of programming languages with a long history and a distinctive, fully parenthesized prefix notation, originally specified in 1958 when John McCarthy developed it at the Massachusetts Institute of Technology (MIT), making it the second-oldest high-level programming language still in common use.

📖 ALGOL (short for "Algorithmic Language") is a family of imperative computer programming languages originally developed in 1958 that heavily influenced many other languages and became the standard method for algorithm description.

🧠 In machine learning, the perceptron (or McCulloch-Pitts neuron) is an algorithm for supervised learning of binary classifiers, with the first implementation being a machine built in 1958 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.

🧠 The term machine learning was coined in 1959 by Arthur Samuel, an IBM employee and pioneer in the field of computer gaming and artificial intelligence. The synonym self-teaching computers was also used in this time period.

1960s

🔑 Passwords have been used with computers since the earliest days of computing. The Compatible Time-Sharing System (CTSS), an operating system introduced at MIT in 1961, was the first computer system to implement password login.

⚙️ In May 1961, IBM engineer Bob Bemer submitted a proposal to the American Standards Association's (ASA) X3.2 subcommittee, initiating formal standardization

work on ASCII (American Standard Code for Information Interchange). Bemser, who also invented the escape sequence, later became known as "the father of ASCII."

📖 Simula is the name of two simulation programming languages developed in the 1960s at the Norwegian Computing Center in Oslo by Ole-Johan Dahl and Kristen Nygaard, with Simula 67 introducing objects, classes, inheritance, subclasses, virtual procedures, coroutines, discrete event simulation, and garbage collection, first appearing in 1962.

🧠 In machine learning, backpropagation is a gradient estimation method used to train neural network models. The term "back-propagating error correction" was introduced in 1962 by Frank Rosenblatt, but he did not know how to implement it, even though Henry J. Kelley had a continuous precursor of backpropagation already in 1960 in the context of control theory.

🎭 Diffusion of innovations is a theory of how, why, and at what rate new ideas and technology spread, developed by sociologist Everett Rogers, who synthesized over 508 diffusion studies into his 1962 book *Diffusion of Innovations*, introducing the classification of adopters into innovators, early adopters, early majority, late majority, and laggards.

⚙️ The first edition of ASCII (ASA X3.4-1963) was published in 1963, coinciding with the introduction of the Teletype Model 33. It was first used commercially as a seven-bit teleprinter code for AT&T's TWX (TeletypeWriter eXchange) network.

💻 In 1964, for the Multics operating system, Louis Pouzin conceived the idea of "using commands somehow like a programming language," and coined the term shell to describe it.

🧠 Expert systems were formally introduced around 1965 by the Stanford Heuristic Programming Project led by Edward Feigenbaum, who is sometimes termed the "father of expert systems," with Dendral — a system designed to identify the molecular structure of unknown organic compounds from mass spectrometry data — being among the earliest, developed with Bruce Buchanan and Joshua Lederberg.

📖 Amdahl's law, a formula limiting the speedup of a task as resources are added to the system executing that task, was presented by Gene Amdahl in his paper "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities" at the AFIPS Spring Joint Computer Conference in April 1967.

🎭 Conway's law is an adage that states organizations design systems that mirror their own communication structure. It is named after the computer programmer Melvin Conway, who introduced the idea in 1967.

 The PIN originated with the introduction of the automated teller machine (ATM) in 1967, as an efficient way for banks to dispense cash to their customers. The first ATM system was that of Barclays in London, in 1967; it accepted cheques with machine-readable encoding, rather than cards, and matched the PIN to the cheque.

 Pseudoterminals appeared as early as 1967 in the DEC PDP-6 Timesharing Monitor, where they were primarily used to implement batch processing.

 The Mother of All Demos, a 90-minute live demonstration by Douglas Engelbart at the ACM/IEEE Fall Joint Computer Conference in San Francisco on December 9, 1968, introduced the oN-Line System (NLS) and demonstrated windows, hypertext, the computer mouse, video conferencing, word processing, and a collaborative real-time editor.

 ed is a line editor for Unix and Unix-like operating systems, developed by Ken Thompson in August 1969 on a PDP-7 at AT&T Bell Labs as one of the first three key elements of the Unix operating system alongside the assembler and shell.

 Hoare logic is a formal system with a set of logical rules for reasoning rigorously about the correctness of computer programs. It was proposed in 1969 by the British computer scientist and logician Tony Hoare, and subsequently refined by Hoare and other researchers.

 Unix (trademarked as UNIX) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, first released in 1969.

 The Advanced Research Projects Agency Network (ARPANET) was the first wide-area packet-switched network with distributed control and one of the first networks to implement the TCP/IP protocol suite, both of which became the technical foundation of the Internet, established in 1969.

 In computing, "server" dates at least to RFC 5 (1969), one of the earliest documents describing ARPANET (the predecessor of Internet), and is contrasted with "user", distinguishing two types of host: "server-host" and "user-host".

 Telnet (short for "teletype network") is a client/server application protocol that provides access to virtual terminals of remote systems on local area networks or the Internet, developed as secret technology beginning in 1969 with RFC 15.

 TENEX was an operating system developed in 1969 by BBN for the PDP-10, which later formed the basis for Digital Equipment Corporation's TOPS-20 operating system. It introduced a full virtual memory system — allowing every program to access an address space of 262,144 words — implemented with a custom BBN-designed hardware pager, and its EXEC command interpreter pioneered command completion,

expanding partial command words into complete words when the user pressed the escape key, a feature still found in modern shells such as tcsh.

1970-74

 The term "relational database" was first defined by E. F. Codd at IBM in 1970. Codd introduced the term in his research paper "A Relational Model of Data for Large Shared Data Banks".

 The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network, introduced on April 16, 1971.

 mail is a command-line email client for Unix and Unix-like operating systems, first released on November 3, 1971.

 roff is a typesetting markup language and the first Unix text-formatting computer program, serving as a predecessor of the nroff and troff document processing systems, first released on November 3, 1971.

 In 1971 the first ARPANET network mail was sent, introducing the now-familiar address syntax with the '@' symbol designating the user's system address. Over a series of RFCs, conventions were refined for sending mail messages over the File Transfer Protocol.

 yacc (Yet Another Compiler Compiler) is a parser generator for context-free grammars that produces C code for LALR parsers. It was first developed by Stephen C. Johnson at Bell Labs in 1971, inspired by Donald Knuth's work on LR parsing; the self-deprecating name reflected the existence of other compiler-compilers at Bell Labs from the Multics project. By 1973 it had reached a form recognizably similar to its C implementation, though early versions were extremely slow and Johnson rewrote it over a dozen times to improve performance by a factor of roughly 10,000.

 ML (Meta Language) was released in 1972 by Robin Milner at Stanford (and later Cambridge) as the meta-language for the LCF (Logic for Computable Functions) proof assistant, providing the foundational type system for OCaml.

 C is a middle-level, general-purpose computer programming language created by Dennis Ritchie between 1972 and 1973 at Bell Labs to construct utilities running on Unix, remaining very widely used and influential due to its design that cleanly reflects the capabilities of targeted CPUs.

 Smalltalk is an object-oriented, dynamically typed reflective programming language designed for educational use as the product of research led by Alan Kay at Xerox Palo

Alto Research Center (PARC), with the first system (Smalltalk-72) running on a Xerox Alto and designed to support Kay's new object-oriented programming paradigm.

 In June 1972, five IBM engineers from the AI department founded SAP (Systemanalyse und Programmentwicklung, "System Analysis and Program Development") as a private partnership under German Civil Code, launching its first commercial product, the RF financial accounting system, in 1973.

 The Xerox Alto, developed at Xerox PARC with first machines introduced on March 1, 1973, was the first computer to demonstrate the desktop metaphor and a bitmapped graphical user interface, later heavily influencing the design of the Macintosh.

 Ethernet, a networking technology for local area networks, was invented by Robert Metcalfe at Xerox PARC and first ran on May 22, 1973, at a bit rate of 2.94 Mbps; Metcalfe named the concept "Ethernet" in a memo written around that time.

 In 1973, Version 4 Unix was rewritten in the higher-level language C, contrary to the general notion at the time that an operating system's complexity and sophistication required it to be written in assembly language.

 sed ("stream editor") is a Unix utility that parses and transforms text using a simple, compact programming language, developed from 1973 to 1974 by Lee E. McMahon of Bell Labs.

 TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network. In May 1974, Vint Cerf and Bob Kahn described an internetworking protocol for sharing resources using packet switching among network nodes, with the specification of the resulting protocol (TCP/IP) written by Vint Cerf, Yogen Dalal, and Carl Sunshine and published in December 1974.

 Structured Query Language (SQL) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS), first appearing in 1974.

 MYCIN is an early backward chaining expert system that uses artificial intelligence to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient's body weight, developed over five or six years in the early 1970s at Stanford University by Edward H. Shortliffe under the direction of Bruce G. Buchanan and others at the Stanford Heuristic Programming Project.

1975-79

 The Data Encryption Standard is a symmetric-key algorithm for the encryption of digital data whose short key length of 56 bits makes it too insecure for modern applications, yet it has been highly influential in the advancement of cryptography. The origins of DES date to 1972 when a National Bureau of Standards study of US government computer security identified a need for a government-wide standard for encrypting unclassified, sensitive information. On March 17, 1975, the proposed DES was published in the Federal Register, with public comments requested and two open workshops held in the following year to discuss the proposed standard.

 *The Mythical Man-Month: Essays on Software Engineering* by Fred Brooks was first published in 1975, drawing on his experience managing IBM's OS/360 project to argue that adding manpower to a software project that is already late makes it later, an idea that became known as Brooks's law.

 The cron command-line utility is a job scheduler on Unix-like operating systems used to schedule jobs (commands or shell scripts), also known as cron jobs, to run periodically at fixed times, dates, or intervals, first released in May 1975.

 lex is a lexical analyzer generator that reads a specification of tokens and produces C code for a lexer. It was developed at Bell Labs by Mike Lesk and Eric Schmidt and first documented in October 1975 as Computing Science Technical Report #39. The same year, Stephen C. Johnson published the formal yacc paper; among the first languages implemented with yacc were AWK, C++, eqn, and Pic.

 Make is a build automation tool that automatically builds executable programs and libraries from source code by reading files called Makefiles that specify how to derive the target program, first appearing in April 1976.

 Metcalfe and Boggs published "Ethernet: Distributed Packet Switching for Local Computer Networks" in *Communications of the ACM* in July 1976.

 vi is a screen-oriented text editor originally created for the Unix operating system, with the original code written by Bill Joy in 1976 as the visual mode for a line editor called ex that Joy had written with Chuck Haley.

 Diffie–Hellman key exchange is a mathematical method of securely exchanging cryptographic keys over a public channel, published in 1976 by Diffie and Hellman as the earliest publicly known work proposing the idea of a private key and a corresponding public key.

 AWK is a domain-specific language designed for text processing and typically used as a data extraction and reporting tool, initially developed in 1977 by Alfred Aho

(author of `egrep`), Peter J. Weinberger (who worked on tiny relational databases), and Brian Kernighan.

 RSA is a public-key cryptosystem widely used for secure data transmission, with the acronym "RSA" coming from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977.

 The Data Encryption Standard was standardized in January 1977, despite its short 56-bit key length making it too insecure for modern applications, yet it has been highly influential in the advancement of cryptography.

 Bill Joy's `ex` 1.1 was released as part of the first Berkeley Software Distribution (BSD) Unix release in March 1978, with many of the ideas in this visual mode taken from Bravo — the bimodal text editor developed at Xerox PARC for the Alto.

 In 1978, Brian Kernighan and Dennis Ritchie published the first edition of *The C Programming Language*. This book, known to C programmers as K&R, served for many years as an informal specification of the language. The version of C that it describes is commonly referred to as "K&R C".

 TeX is a typesetting system designed and written by Donald Knuth and first released in 1978. The first version of TeX, called TeX78, was written in the SAIL programming language to run on a PDP-10 under Stanford's WAITS operating system.

 Taiichi Ohno published his seminal book, *Toyota Production System: Beyond Large-Scale Production*, in 1978, formalizing the philosophy for a global audience.

 The problem of obtaining Byzantine consensus was conceived and formalized by Robert Shostak, who dubbed it the interactive consistency problem. This work was done in 1978 in the context of the NASA-sponsored SIFT project in the Computer Science Lab at SRI International.

 The C shell (`csh`) is a Unix shell created by Bill Joy while he was a graduate student at the University of California, Berkeley in the late 1970s, and distributed beginning with the 2BSD release of the Berkeley Software Distribution which Joy first distributed in 1978. Its syntax is modelled on the C programming language, and other early contributors to the ideas or the code were Michael Ubell, Eric Allman, Mike O'Brien and Jim Kulp.

 Model-view-controller (MVC) is a software design pattern commonly used for developing user interfaces that divides the related program logic into three interconnected elements. Trygve Reenskaug created MVC while working on Smalltalk-79 as a visiting scientist at the Xerox Palo Alto Research Center (PARC) in the late 1970s.

 The Bourne shell, `sh`, was a new Unix shell developed by Stephen Bourne at Bell

Labs and distributed as the shell for UNIX Version 7 in 1979.

⚙️ A chroot on Unix and Unix-like operating systems is an operation that changes the apparent root directory for the current running process and its children. The chroot system call was introduced during development of Version 7 Unix in 1979.

📖 Both lex and yacc became standard tools in the Unix toolchain with the release of Version 7 Unix in 1979, establishing them as foundational components for compiler and language development on Unix systems.

🏢 Oracle Database (commonly referred to as Oracle DBMS, Oracle Autonomous Database, or simply Oracle) is a proprietary multi-model database management system produced and marketed by Oracle Corporation, first released in 1979.

⚙️ Jef Raskin, an Apple employee, initiated the Macintosh project in 1979, envisioning an affordable, easy-to-use computer for the masses. He named the project after his favorite apple variety, the McIntosh.

🧠 An independent evaluation of MYCIN was conducted in 1979 at Stanford Medical School, in which the system received an acceptability rating of 65%, comparable to or exceeding the 42.5% to 62.5% ratings given to faculty physicians, demonstrating that rule-based expert systems could achieve performance on par with human specialists in targeted diagnostic tasks.

🧠 Prospect theory is a theory of behavioral economics, judgment, and decision-making that describes how people choose between probabilistic alternatives, introduced in a 1979 *Econometrica* paper, "Prospect Theory: An Analysis of Decision under Risk," by Daniel Kahneman and Amos Tversky as a descriptive alternative to expected utility theory.

🌐 Usenet, a worldwide distributed discussion system, was conceived in 1979 when Tom Truscott and Jim Ellis of Duke University came up with the idea as a replacement for a local announcement program, with the public release of the news software written by Steve Daniel and Truscott.

1980-84

⚙️ Digital Equipment Corporation, Intel, and Xerox (DIX) published version 1.0 of "The Ethernet: A Local Area Network — Data Link Layer and Physical Layer Specifications," known as the "Blue Book," on September 30, 1980.

📖 Smalltalk-80 was the first language variant made available outside of PARC, first as Smalltalk-80 Version 1, given to a small number of firms and universities.

🔗 Usenet was connected to ARPANET through UC Berkeley in 1980, marking its initial link to the broader network beyond the original UUCP dial-up connections.

⚙️ Steve Jobs took over the Macintosh project in 1981 after being removed from the Lisa team, redirecting the design toward a Xerox PARC-inspired graphical user interface while retaining Jef Raskin's vision of producing computers by the millions.

⚙️ The Berkeley r-commands are a suite of computer programs designed to enable users of one Unix system to log in or issue commands to another Unix computer via TCP/IP network, first released in June 1981.

🔗 By June 1, 1981, the UUCP/Usenet network had expanded to include dozens of connected systems across universities and research institutions.

⚙️ IPv4 is described in RFC 791 (1981).

💻 tcsh, an enhanced C shell, has its origins in Ken Greer's work at Carnegie Mellon University to implement TENEX-style filename completion, which he began in September 1975 and finally merged into the C shell in December 1981. The "t" in tcsh comes from the "T" in TENEX, the operating system whose command-completion feature inspired Greer.

⚙️ In March 1982, the US Department of Defense declared TCP/IP as the standard for all military computer networking.

📅 *The Mythical Man-Month* was reprinted with corrections in 1982, by which point Brooks's law had become a foundational principle in software project management

📊 Microsoft Multiplan, an early spreadsheet program developed by Microsoft, was released in August 1982.

📖 Revision Control System (RCS) is an early implementation of a version control system (VCS), a set of UNIX commands that allow multiple users to develop and maintain program code or documents. RCS was first released in 1982 by Walter F. Tichy at Purdue University and is currently maintained by the GNU Project.

💻 John Warnock and Charles Geschke left Xerox PARC and founded Adobe Systems in December 1982. They began developing PostScript, a page description language drawing on Warnock's earlier Interpress research at Xerox, with the goal of providing a device-independent way to describe printed pages.

📖 TeX82, a new version of TeX rewritten from scratch, was published in 1982. Among other changes, the original hyphenation algorithm was replaced by a new algorithm written by Frank Liang.

⚙️ The migration of the ARPANET from NCP to TCP/IP was officially completed on flag day January 1, 1983, when the new protocols were permanently activated.

📊 Lotus 1-2-3 was officially released for the IBM PC on January 26, 1983, quickly becoming the platform's "killer app" by integrating spreadsheet, graphing, and data management capabilities into a single, high-performance program written in x86 assembly.

💻 KornShell (ksh) is a Unix shell developed by David Korn at Bell Labs in the early 1980s and announced at USENIX on July 14, 1983. It is backward-compatible with the Bourne shell and includes many features of the C shell, inspired by the requests of Bell Labs users.

⚙️ The r-commands (including rlogin, rsh, and rcp) were fully incorporated into 4.2BSD in August 1983, a release that provided the mature interprocess communication (IPC) primitives that enabled them to become the de facto standards for Unix networking throughout the 1980s.

📖 Development of the GNU operating system was initiated by Richard Stallman while he worked at MIT Artificial Intelligence Laboratory. It was called the GNU Project, and was publicly announced on September 27, 1983.

⚙️ The Internet Engineering Task Force published the DNS original specifications in RFC 882 and RFC 883 in November 1983.

⚙️ The Simple Mail Transfer Protocol (SMTP) protocol was implemented on the ARPANET in 1983.

⚙️ The modern Unix pseudoterminal interface originated in 1983 during the development of Eighth Edition Unix and was widely popularized following its inclusion in the 4.2BSD (Berkeley Software Distribution) release.

⚙️ In 1983, IEEE published the 802.3 standard as a draft for 10 Mbps Ethernet over coaxial cable up to 500 meters (10BASE5).

💻 In the spring of 1983, Steve Jobs visited Adobe and was dazzled by PostScript's potential for the Macintosh. Apple and Adobe signed a PostScript licensing deal in December 1983, with Apple investing in Adobe and agreeing to use PostScript in its upcoming laser printer.

⚙️ The original Macintosh 128K was introduced by Steve Jobs on January 24, 1984, at a price of US\$2,495. It was the first commercially successful personal computer to feature a mouse-driven graphical user interface, using a Motorola 68000 processor and a 9-inch monochrome display.

⚙️ In 1984, four UC Berkeley students, Douglas Terry, Mark Painter, David Riggle, and Songnian Zhou, wrote the first Unix name server implementation for the Berkeley Internet Name Domain, commonly referred to as BIND.

🏢 X/Open group was a consortium founded by several European UNIX systems manufacturers in 1984 to identify and promote open standards in the field of information technology.

🏢 Eliyahu M. Goldratt introduced the Theory of Constraints to a wide audience in his business novel, *The Goal*, in 1984.

📖 TeX has been the official typesetting package for the GNU operating system since 1984.

🧠 The seminal book *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* was published in 1984 by Bruce Buchanan and Edward Shortliffe, documenting the MYCIN system and EMYCIN (Essential MYCIN) — an expert system shell stripped of its medical knowledge base — which enabled construction of rule-based expert systems across diverse domains and contributed to the rapid proliferation of expert systems throughout the 1980s, when two-thirds of Fortune 500 companies applied the technology in daily business activities.

1985

💻 Apple and Adobe announced the LaserWriter on January 23, 1985 — the first printer to ship with PostScript — at Apple's annual shareholder meeting. Aldus simultaneously announced PageMaker, triggering the desktop publishing revolution.

📖 GNU Emacs is a free software text editor created by GNU Project founder Richard Stallman based on the Emacs editor developed for Unix operating systems, written in C and providing Emacs Lisp (also implemented in C) as an extension language, with version 13 as the first public release on March 20, 1985.

🏢 The GNU Manifesto is a call-to-action by Richard Stallman encouraging participation and support of the GNU Project's goal in developing the GNU free computer operating system. It was published in March 1985.

💻 Aldus PageMaker 1.0 was released in July 1985 for the Macintosh with native PostScript support, becoming the first widely adopted desktop publishing application and cementing PostScript as the standard page description language for professional printing. PageMaker won a Codie Award for Best New Use of a Computer in 1986.

📊 Microsoft Excel was first released for the Macintosh on September 30, 1985, and was the first spreadsheet to allow the user to define the appearance of spreadsheets.

📊 Lotus 1-2-3 Release 2.0 was launched in September 1985, introducing support for macros, add-ins, and expanded memory (EMS), which solidified its dominance in the corporate spreadsheet market.

⚙️ Microsoft Windows was first announced by Bill Gates on November 10, 1983, as a graphical user interface for MS-DOS, and Windows 1.0 was officially released on November 20, 1985.

📖 LaTeX was created in the early 1980s by Leslie Lamport while working at Stanford Research Institute (SRI), originally written to fulfill his own need for TeX macros but designed with the intent to be made into a general package usable by others, with versions released in 1984 and 1985.

🏢 The Free Software Foundation was founded in 1985 as a non-profit corporation supporting free software development.

📖 C is a high-level general-purpose programming language created by Danish computer scientist Bjarne Stroustrup as an extension of the C programming language (or "C with Classes"), first appearing in 1985 and initially standardized in 1998 as ISO/IEC 14882:1998, which was subsequently amended by the C03, C11, C14, and C++17 standards.

📖 GNU Bison, a free yacc-compatible parser generator, was originally written by Robert Corbett in 1985. Richard Stallman subsequently made it fully yacc-compatible as part of the GNU Project, with Wilfred Hansen of Carnegie Mellon University adding multi-character string literals and other features.

1986

🏢 Hirotaka Takeuchi and Ikujiro Nonaka introduced the term "scrum" in the context of product development in their Harvard Business Review article "The New New Product Development Game" in January–February 1986.

📖 GDB is free software released under the GNU General Public License, first written by Richard Stallman in 1986 as part of his GNU system after GNU Emacs reached a "reasonably stable" state.

🌐 The Standard Generalized Markup Language (SGML; ISO 8879:1986) is a standard for defining generalized markup languages for documents.

📖 Dick Grune at Vrije Universiteit Amsterdam developed an early form of CVS in July 1986 as a set of shell scripts wrapping RCS, to allow multiple developers to work on the same files concurrently.

🧠 In machine learning, backpropagation is a widely used algorithm for training feedforward artificial neural networks or other parameterized networks with differentiable nodes, with David E. Rumelhart et al. publishing an experimental analysis in 1986 that contributed to its popularization and helped initiate an active period of research in multilayer perceptrons.

🧠 The term Deep Learning was introduced to the machine learning community by Rina Dechter in 1986.

📊 gnuplot is a command-line and GUI program that can generate two- and three-dimensional plots of functions, data, and data fits across all major computers and operating systems (Linux, Unix, Microsoft Windows, macOS, FreeDOS, and many others), first released in 1986.

📊 The POSTGRES project officially launched at UC Berkeley in 1986 under the leadership of Michael Stonebraker, aiming to solve the limitations of the relational model by introducing object-relational concepts.

🔧 RFC 977, published in 1986, provided the Network News Transfer Protocol (NNTP) specification for distributing Usenet articles over TCP/IP.

1987

⚙️ The Macintosh SE and Macintosh II were both introduced at the AppleWorld conference in Los Angeles on March 2, 1987. The Macintosh II was Apple's first modular and color-capable Macintosh, breaking from the all-in-one design of earlier models.

📖 The GNU Compiler Collection (GCC) is an optimizing compiler produced by the GNU Project supporting various programming languages, hardware architectures, and operating systems. Originally named the GNU C Compiler when first released by Richard Stallman on March 22, 1987 (available via FTP from MIT) because it only supported the C programming language, GCC has since evolved to support many more languages.

📖 "No Silver Bullet — Essence and Accidents of Software Engineering" is an essay by Fred Brooks arguing that no single development in technology or management technique promises an order-of-magnitude improvement in software productivity within a decade, distinguishing "essential" complexity inherent to a problem from "accidental" complexity introduced by tools and processes; first presented at the 1986 IFIP Tenth World Computing Conference and republished in *IEEE Computer* in April 1987.

📖 flex (Fast Lexical Analyzer Generator) is a free reimplement of lex that produces faster and more efficient C code for scanners. It was written by Vern Paxson

at UC Berkeley around 1987 with inspiration from Van Jacobson, removing the proprietary licensing restrictions that had limited the original AT&T lex's adoption outside commercial Unix environments.

🧠 *Peopleware: Productive Projects and Teams* by Tom DeMarco and Tim Lister was first published in 1987.

📖 In September 1987, a meeting at the conference on Functional Programming Languages and Computer Architecture (FPCA) in Portland, Oregon, reached a strong consensus that a committee should be formed to design a common non-strict, purely functional language, as more than a dozen such languages of similar expressive power had come into being. The committee named the language Haskell, in honour of the logician Haskell B. Curry.

⚙️ Windows 2.0, released on December 9, 1987, introduced overlapping windows, desktop icons, and keyboard shortcuts, significantly improving the user experience over the previous version.

📖 Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier, first appearing on December 18, 1987.

📊 The first Windows version of Excel was released on November 19, 1987, numbered as version 2.0 to be in line with the Macintosh version, and included a run-time version of Windows.

📖 Self is an object-oriented programming language based on the concept of prototypes, designed mostly by David Ungar and Randall Smith while working at Xerox PARC in 1986. It began as a Smalltalk dialect with dynamic typing, just-in-time compilation (JIT), and a prototype-based approach to objects, first appearing in 1987.

📖 The first implementation of Caml (Categorical Abstract Machine Language) was created by Ascánder Suárez at INRIA in 1987 based on the Categorical Abstract Machine compiling method.

📊 SQL was adopted as a standard by the ANSI in 1986 as SQL-86 and the ISO in 1987.

1988

🧠 Cognitive load theory is an instructional theory that differentiates cognitive load into intrinsic, extraneous, and germane load, developed by educational psychologist John Sweller, who published "Cognitive Load During Problem Solving: Effects on Learning" in *Cognitive Science* in April 1988, showing that means-ends problem-solving strategies can impose cognitive demands that hinder schema acquisition.

📊 Wolfram Mathematica is a software system with built-in libraries for technical computing including machine learning, statistics, symbolic computation, data manipulation, network analysis, time series analysis, NLP, optimization, plotting, algorithm implementation, user interface creation, and interfacing with programs in other languages, first released on June 23, 1988.

🔗 Joe Becker of Xerox published a draft proposal in August 1988 for an international/multilingual text character encoding system tentatively called "Unicode," outlined in a document entitled *Unicode 88*, proposing a 16-bit character scheme developed with Lee Collins and Mark Davis of Apple.

🔗 Internet Relay Chat (IRC), a text-based chat system, was created by Jarkko Oikarinen in August 1988 at the University of Oulu, Finland, originally intended to replace a program called MUT (MultiUser Talk) on a BBS called OuluBox.

🐛 The Morris worm or Internet worm of November 2, 1988, is one of the oldest computer worms distributed via the Internet, and the first to gain significant mainstream media attention.

🔒 X.509 is an International Telecommunication Union (ITU) standard defining the format of public key certificates, first published on November 25, 1988.

📖 GNU Make (short gmake) is the standard implementation of Make for Linux and macOS, providing several extensions over the original Make such as conditionals, built-in functions to eliminate the need for shell scripting in makefile rules, and the ability to manipulate variables, first released in 1988.

📊 GNU Octave was conceived in 1988 by John W. Eaton at the University of Texas at Austin as a high-level language for numerical computations, originally intended as a companion for a chemical reactor design course.

📖 AWK was significantly revised and expanded in 1985-88, resulting in the GNU AWK implementation written by Paul Rubin, Jay Fenlason, and Richard Stallman, released in 1988.

📊 The Open Software Foundation (OSF) was a not-for-profit industry consortium formed in 1988 to create an open standard for Unix operating system implementations.

📖 The Portable Operating System Interface (POSIX) is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems, defining both system- and user-level application programming interfaces (APIs), developed starting in 1988.

📊 The concept of data warehousing dates back to the late 1980s when IBM

researchers Barry Devlin and Paul Murphy developed the "business data warehouse" and published the article "An architecture for a business and information system" in 1988 introducing the term to the broader audience.

🧠 The Design of Everyday Things is a book by cognitive scientist Donald Norman, originally published in 1988 under the title *The Psychology of Everyday Things* (POET), that describes design principles rooted in cognitive psychology and advocates for user-centered design based on human needs and cognition, with a revised edition released in 2013.

💻 The functionality of the original KornShell, ksh88, was used as a basis for the POSIX.2 standard. The ksh88 version, named after 1988, became the default shell on AIX and several other commercial Unix systems.

1989

🔑 Kerberos is a computer-network authentication protocol that works using tickets to allow nodes communicating over non-secure networks to prove their identity to one another securely, developed by the Massachusetts Institute of Technology (MIT) in 1988 to protect network services provided by Project Athena. Kerberos version 4, the first public version, was released on January 24, 1989.

📊 Microsoft SQL Server is a proprietary relational database management system developed by Microsoft that functions as a database server with the primary purpose of storing and retrieving data as requested by other software applications, with initial release on April 24, 1989.

📄 Brian Berliner redesigned and rewrote CVS in C in April 1989, vastly improving performance and reliability over Grune's original shell script implementation.

💻 Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell, first released on June 8, 1989, and has been used as the default login shell for most Linux distributions.

📊 Lotus 1-2-3 Release 3.0 was released in June 1989 as a major rewrite in the C language, introducing "3D spreadsheets" with multiple worksheets in a single file, though it required newer 80286-based hardware to run.

📊 POSTGRES Version 1 was released to a small group of external users in June 1989, marking the first public implementation of the extensible database system developed at Berkeley.

⚙️ NeXTSTEP 1.0, the object-oriented, multitasking operating system developed by NeXT Computer founded by Steve Jobs after leaving Apple in 1985, was released on

September 18, 1989. Built on the Mach kernel and BSD, it would later become the foundation for Mac OS X and modern macOS.

🌐 The Hypertext Transfer Protocol (HTTP) is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems. Development of HTTP was initiated by Tim Berners-Lee at CERN in 1989 and summarized in a simple document describing the behavior of a client and a server using the first HTTP protocol version that was named 0.9.

Timeline - 1990-99

1990

📖 The Haskell version 1.0 Report, the first definition of the language, was published on April 1, 1990, edited by Paul Hudak and Philip Wadler. The Haskell mailing list was started at the same time, open to all.

⚙️ Windows 3.0, launched on May 22, 1990, became the first version of Windows to achieve widespread commercial success, featuring a completely redesigned user interface and improved memory management for 386 processors.

📖 groff (also called GNU troff) is a typesetting system that creates formatted output when given plain text mixed with formatting commands, with version 0.3.1 first released in June 1990.

📖 CVS (Concurrent Versions System) version 1.0 was publicly released on November 19, 1990, becoming one of the dominant version control systems throughout the 1990s.

💻 Zsh is a Unix shell that can be used as an interactive login shell and as a command interpreter for shell scripting. Paul Falstad wrote the first version in 1990 while a student at Princeton University, posting it to the alt.sources Usenet newsgroup in eight parts on December 14, 1990. The name derives from Zhong Shao, a teaching assistant at Princeton, whose login "zsh" Falstad regarded as a good name for a shell. By version 1.0 the aim was to be a cross between ksh and tcsh, a powerful "command and programming language" that is well-designed and logical like ksh, but also built for humans like tcsh.

🌐 CERN httpd (later also known as W3C httpd) is an early, now discontinued, web server (HTTP) daemon originally developed at CERN from 1990 onwards by Tim Berners-Lee, Ari Luotonen and Henrik Frystyk Nielsen. Implemented in C, it was the first web server software, with its initial release on December 24, 1990.

📖 The book "The Machine That Changed the World" was published in 1990, coining

the term "Lean Manufacturing" to describe the Toyota Production System methodology to the West.

⚙️ Microsoft released OLE 1.0 (Object Linking and Embedding) in 1990, introducing the first document-linking and embedding technology for Windows that allowed software components to interact.

📖 flex was publicly released in 1990 under a BSD-style license by the Regents of the University of California, making it the first freely available replacement for the proprietary AT&T lex and enabling widespread adoption across open-source Unix projects.

🌐 In 1990, Tim Berners-Lee's proposals for hypertext implicitly introduced the idea of a URL as a short string representing a resource that is the target of a hyperlink.

1991

✂️ The Unicode Consortium was incorporated in California on January 3, 1991, as a non-profit organization to develop, extend, and promote the use of the Unicode Standard.

🌐 The first web browser, WorldWideWeb, was developed in 1990 by Tim Berners-Lee for the NeXT Computer and introduced to his colleagues at CERN in March 1991.

⚙️ The Linux kernel is a free and open-source, monolithic, modular, multitasking, Unix-like operating system kernel. In April 1991, Linus Torvalds, a 21-year-old computer science student at the University of Helsinki, started working on a simple operating system inspired by UNIX for his i386-based PC, beginning with a task switcher in Intel 80386 assembly language and a terminal driver.

⚙️ Linus Torvalds announced his project on the [comp.os.minix](#) newsgroup on August 25, 1991, famously describing it as "just a hobby, won't be big and professional like gnu."

⚙️ On 17 September 1991, Torvalds released version 0.01 of the Linux kernel on the FUNET FTP server. It was not yet executable and required MINIX for compilation.

⚙️ On 5 October 1991, Torvalds announced the first "official" version of Linux, version 0.02. At this point, the kernel was capable of running the Bash shell and GCC.

✂️ Unicode 1.0 was published in October 1991, encoding 7,161 characters from 24 scripts in a unified 16-bit character set intended to encompass all modern writing systems.

📖 Vim (a contraction of Vi IMproved) is a free and open-source, screen-based text

editor program and an improved clone of Bill Joy's vi. Vim's author, Bram Moolenaar, derived Vim from a port of the Stevie editor for Amiga and released the initial version to the public on November 2, 1991.

 Python is a high-level, interpreted, general-purpose programming language with design philosophy emphasizing code readability through significant indentation. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.

 In 1991, the autoencoder was first proposed as a nonlinear generalization of principal components analysis (PCA) by Kramer.

 Pretty Good Privacy (PGP) is an encryption program that provides cryptographic privacy and authentication for data communication. PGP is used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. Phil Zimmermann developed PGP in 1991.

 In 1991, the rlogin protocol was formally defined in RFC 1282, documenting the existing practice that had been widely adopted by the Unix community for over a decade.

 Crossing the Chasm is a marketing book by Geoffrey A. Moore, first published in 1991, that applies Everett Rogers' diffusion of innovations theory specifically to technology products, describing a "chasm" between early adopters and the early majority that new technologies must cross to achieve mainstream success, with revised editions following in 1999 and 2014.

1992

 Windows 3.1, released on April 6, 1992, introduced TrueType fonts, the Windows Registry, and dropped support for Real Mode, marking a major step toward a more modern operating system architecture.

 Linux 0.12 was released on February 1, 1992, under the GNU General Public License (GPL), marking a pivotal shift from its original restrictive license.

 MD5 is one in a series of message digest algorithms designed by Professor Ronald Rivest of MIT. When analytic work indicated that MD5's predecessor MD4 was likely to be insecure, Rivest designed MD5 in 1991 as a secure replacement and first published it in April 1992.

 UTF-8 was designed by Ken Thompson and Rob Pike on September 2, 1992, sketched on a placemat in a New Jersey diner. They implemented it and updated Plan 9

from Bell Labs to use it throughout, creating a self-synchronizing, backward-compatible variable-length encoding for Unicode.

🌐 libwww (Library World Wide Web) is a modular client-side web API for Unix and Windows, and the reference implementation of the libwww API. In 1991 and 1992, Tim Berners-Lee and a student at CERN named Jean-François Groff rewrote various components of the original WorldWideWeb browser for the NeXTstep operating system in portable C code to demonstrate the potential of the World Wide Web, with version 1.0 initially released in November 1992.

🏢 In November 1992 the IETF "URI Working Group" met for the first time.

📖 CTAN (an acronym for "Comprehensive TeX Archive Network") is the authoritative place where TeX related material and software can be found for download. Built in 1992 by Rainer Schöpf and Joachim Schrod in Germany, Sebastian Rahtz in the UK, and George Greenwade in the U.S., it was officially announced at the EuroTeX conference at Aston University in 1993. The WEB server itself is maintained by Gerd Neugebauer.

🏢 The term technical debt is a qualitative description of the cost to maintain a system that is attributable to choosing an expedient solution for its development. The term was coined by Ward Cunningham in 1992. After reading *Metaphors We Live By*, Ward devised this debt metaphor to explain to his boss the need to refactor the financial product they were working on.

⚙️ The original Berkeley Packet Filter (BPF) was introduced in 1992 in a research paper by Steven McCanne and Van Jacobson. It provided a way to filter network packets in the kernel, minimizing data copying between the kernel and user space.

1993

🌐 NCSA Mosaic 0.5, the first alpha/beta version of the web browser developed at the National Center for Supercomputing Applications (NCSA), was announced by Marc Andreessen for the X Window System on January 23, 1993.

📖 PDF (Portable Document Format) was publicly introduced by Adobe Systems in January 1993 at the Windows and OS/2 Conference. Adobe made the PDF 1.0 specification freely available and released Acrobat 1.0 as the first reader/writer, aiming to enable consistent document exchange across any platform.

⚙️ The NetBSD source code repository was established on March 21, 1993. The project was founded by Chris Demetriou, Theo de Raadt, Adam Glass, and Charles Hannum as a more open and community-driven alternative to 386BSD.

🌐 NCSA Mosaic 1.0 for the X Window System was officially released on April 21, 1993.

It was the first browser to display images inline with text using the `<img>` tag, rather than in a separate window, which was instrumental in popularizing the World Wide Web.

⚙️ The name FreeBSD was officially chosen by David Greenman on June 19, 1993. The project began as an effort to maintain the "unofficial 386BSD patchkit" after 386BSD development slowed earlier in the year.

📊 R was started by professors Ross Ihaka and Robert Gentleman as a programming language to teach introductory statistics at the University of Auckland, first appearing in August 1993.

🌐 NCSA released the first versions of Mosaic for Microsoft Windows and Apple Macintosh in September 1993, making the web accessible to non-technical home users.

⚙️ Debian, also known as Debian GNU/Linux, is a Linux distribution composed of free and open-source software, developed by the community-supported Debian Project. The first version of Debian (0.01) was released on September 15, 1993.

⚙️ FreeBSD 1.0-RELEASE was published on November 1, 1993, based on 4.3BSD-Net/2 and 386BSD. This initial release provided a robust BSD-based operating system for the PC architecture.

🌐 NCSA Mosaic 2.0 for Unix was released on November 10, 1993. It introduced support for HTML forms, which enabled the creation of the first dynamic and interactive web pages.

⚙️ The Component Object Model (COM) was officially introduced in 1993 as the underlying binary architecture for OLE 2.0, establishing a language-independent standard for software component interoperability.

⚙️ In 1993, Microsoft introduced OLE Automation, which enabled applications such as Microsoft Excel to expose their functionality to external scripting languages like Visual Basic.

📊 Microsoft released Excel 5.0 in 1993, which included Visual Basic for Applications (VBA), a programming language that added the ability to automate tasks and provide user-defined functions.

⚙️ In 1993, Microsoft launched Windows NT 3.1, the first version of the "New Technology" (NT) line, which featured a robust 32-bit kernel designed for high-end workstations and servers, separate from the consumer-oriented DOS-based line.

⚙️ CFEngine is an open-source configuration management system written by Mark Burgess that provides automated configuration and maintenance of large-scale

computer systems. The CFEngine project began in 1993 as a way for author Mark Burgess to get his work done by automating the management of a small group of workstations in the Department of Theoretical Physics.

🌐 The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser, with its initial release in 1993.

🌐 FutureWave Software (founded by Charlie Jackson, Jonathan Gay, and Michelle Alsip-Welsh) published SmartSketch in 1993, a vector drawing application for pen computers. This was the earliest precursor to what would become Adobe Flash.

🌐 Common Gateway Interface (CGI) is an interface specification that enables web servers to execute an external program, typically to process user requests. In 1993, the National Center for Supercomputing Applications (NCSA) team wrote the specification for calling command line executables on the www-talk mailing list.

🌐 NCSA HTTPd is an early, now discontinued, web server originally developed at the NCSA at the University of Illinois at Urbana-Champaign by Robert McCool and others, first released in 1993 and among the earliest web servers developed.

📖 Lua is a lightweight, high-level, multi-paradigm scripting language designed primarily for embedded use in applications, created in 1993 by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes at Tecgraf, the Computer Graphics Technology Group of the Pontifical Catholic University of Rio de Janeiro in Brazil. Its name means "Moon" in Portuguese.

🧑 Don Norman coined the term "user experience" in 1993 after joining Apple as an Apple Fellow and forming the User Experience Architect's Office, explaining he wanted a term broader than "human interface" or "usability" to cover industrial design, graphics, the interface, and physical interaction.

💻 The ksh93 version of KornShell, released in 1993, added features such as associative arrays and built-in floating-point arithmetic operations, along with dynamically loaded built-in commands and other enhancements over ksh88.

1994

📊 GNU Octave 1.0 was officially released on February 17, 1994, establishing itself as a robust tool for scientific computing and engineering within the GNU project.

⚙️ Linux version 0.95 was the first to be capable of running the X Window System. Linux 1.0.0 was released on March 14, 1994, consisting of 176,250 lines of code. It was the first version suitable for use in production environments.

📄 In June 1994, the IETF published Berners-Lee's first Request for Comments that acknowledged the existence of URLs and URNs.

📄 The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web, founded on October 1, 1994, and led by Tim Berners-Lee.

📄 Perl 5.000 was released on October 17, 1994. It was a nearly complete rewrite of the interpreter, and it added many new features to the language, including objects, references, lexical (my) variables, and modules

🌐 Netscape Navigator was a proprietary web browser and the original browser of the Netscape line (versions 1 to 4.08, and 9.x), serving as the flagship product of Netscape Communications Corp and the dominant web browser in terms of usage share in the 1990s, with initial release on December 15, 1994.

🔲 The QR code system was invented in 1994, at the Denso Wave automotive products company, in Japan.

📄 SUnit, the first unit testing framework, was created by Kent Beck for Smalltalk in 1994. It introduced the test fixture pattern — setUp, test methods, and tearDown — that underlies the entire xUnit family of frameworks.

👤 Jakob Nielsen and Rolf Molich developed a set of usability heuristics for user interface design, with an early version appearing in papers published in 1989–1990 and Nielsen publishing a refined set of ten heuristics in 1994 covering principles such as visibility of system status, error prevention, and consistency, which became a widely used framework for heuristic evaluation in usability engineering.

1995

🔒 Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a computer network. Netscape developed the original SSL protocols, and Taher Elgamal, chief scientist at Netscape Communications from 1995 to 1998, has been described as the "father of SSL". SSL Version 2.0, after being released in February 1995 was quickly discovered to contain a number of security and usability flaws.

📄 Wirth's law is an adage on computer performance which states that software is getting slower more rapidly than hardware is becoming faster, named after Niklaus Wirth, who discussed it in his February 1995 article "A Plea for Lean Software" published in *Computer* magazine.

📄 Java is a high-level, class-based, object-oriented programming language designed to

have as few implementation dependencies as possible. Originally developed by James Gosling at Sun Microsystems, Java SE defines a range of general-purpose APIs and includes the Java Language Specification and the Java Virtual Machine Specification, first appearing on May 23, 1995.

🌐 Java Applets were introduced with Java 1.0a2 and the HotJava browser, publicly demonstrated at SunWorld on May 23, 1995. They allowed small Java programs to be embedded in web pages and executed inside a sandboxed JVM provided by a browser plugin (NPAPI), enabling interactive content on the web.

🗄️ MySQL is an open-source relational database management system (RDBMS) available as free and open-source software under the terms of the GNU General Public License, and also available under a variety of proprietary licenses, with initial release on May 23, 1995.

🌐 PHP is a general-purpose scripting language geared toward web development, originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994 and first appearing on June 8, 1995.

⚙️ Windows 95, released on August 24, 1995, was a massive consumer release that introduced the iconic Start menu, the Taskbar, and Plug and Play technology, while moving the consumer line toward a 32-bit architecture.

⚙️ Theo de Raadt created the OpenBSD CVS repository on October 18, 1995, marking the official start of the project as a fork of NetBSD. The project was established following disagreements between de Raadt and the NetBSD core team, with a new focus on proactive security and code correctness.

📖 CPAN was conceived in 1993 and has been active online since October 1995, based on the CTAN model and beginning as a place to unify the structure of scattered Perl archives. On October 26, 1995, the Comprehensive Perl Archive Network (CPAN) was established as a repository for the Perl language and Perl modules.

📖 Ruby is an interpreted, high-level, general-purpose programming language which supports multiple programming paradigms. The first public release of Ruby 0.95 was announced on Japanese domestic newsgroups on December 21, 1995.

🌐 In December 1995, Sun Microsystems and Netscape announced JavaScript in a press release. The first JavaScript engine was created by Brendan Eich in 1995 for the Netscape Navigator web browser. It was a rudimentary interpreter for the nascent language Eich invented.

⚙️ The first RFC to standardize IPv6 was RFC 1883 in 1995, formally introducing the protocol in December 1995 as the successor to IPv4, expanding the address space from

32 to 128 bits.

 The predecessor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers, with initial release in 1995.

 Fast Ethernet, running at 100 Mbps, was introduced in 1995 as the IEEE 802.3u standard.

 The Apache HTTP Server is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0, developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, with initial release in 1995.

 SHA-1 (Secure Hash Algorithm 1) is a hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest — typically rendered as 40 hexadecimal digits. Designed by the United States National Security Agency, it is a U.S. Federal Information Processing Standard first published in 1995.

 SSH was designed as a replacement for Telnet and for unsecured remote shell protocols such as the Berkeley rsh and the related rlogin and rexec protocols. In 1995, Tatu Ylönen, a researcher at Helsinki University of Technology, Finland, designed the first version of the protocol (now called SSH-1), prompted by a password-sniffing attack at his university network.

 *The Mythical Man-Month* was republished in a 20th-anniversary edition with four extra chapters in 1995, reprinting Brooks's 1986 essay "No Silver Bullet" alongside his new commentary "'No Silver Bullet' Refined."

 Ken Schwaber and Jeff Sutherland formally presented the Scrum framework at the OOPSLA '95 conference in Austin, Texas, in 1995.

1996

 UTF-8 was first formally standardized in January 1996 as RFC 2044, defining it as a transformation format for encoding Unicode characters for use in MIME and Internet protocols while preserving US-ASCII compatibility.

 Netscape Navigator 2.0 was released in March 1996 with Java Applet support via the NPAPI plugin architecture, making applets broadly accessible and sparking their rapid adoption across the web.

 Debian first stable version (1.1) was released on June 17, 1996.

 PostgreSQL, also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. It was

first released as an open-source project on July 8, 1996, and on October 22, 1996, the website PostgreSQL.org was launched as the project was officially renamed to reflect its SQL support.

🌐 Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML, with initial release on December 17, 1996.

🌐 In December 1996, Macromedia acquired FutureWave Software and rebranded FutureSplash Animator as Macromedia Flash 1.0, distributing Flash Player as a free browser plugin to rapidly gain market share and establish Flash as the dominant platform for interactive web content.

🌐 A Jakarta Servlet (formerly Java Servlet) is a Java software component that extends the capabilities of a server, most commonly to handle requests and responses for web applications. The Servlet API was first publicly announced at the inaugural JavaOne conference in May 1996, and Servlet 1.0 shipped in December 1996 as part of the Java Servlet Development Kit, with the first specification created by Pavni Diwanji at Sun Microsystems.

🏢 The Open Group is a global consortium that seeks to "enable the achievement of business objectives" by developing "open, vendor-neutral technology standards and certifications." It was established in 1996 when X/Open merged with the Open Software Foundation.

📄 Objective Caml (OCaml) was first released in 1996, when Didier Rémy and Jérôme Vouillon integrated a powerful, statically type-safe object and class system into Caml Special Light, giving the language its "Objective" prefix.

💻 IntelliSense is Microsoft's implementation of code completion, best known in Visual Studio. It was first introduced as a feature of a mainstream Microsoft product in 1996 building on many already invented concepts of code completion and syntax checking.

🌐 Microsoft launched ActiveX in 1996 as a rebranded and simplified set of COM-based technologies specifically optimized for the web and integration with Internet Explorer.

🌐 HTTP/1 was finalized and fully documented (as version 1.0) in 1996.

🌐 In 1996, the iframe tag was introduced by Internet Explorer; like the object element, it can load or fetch content asynchronously.

🔒 Newer versions of SSL/TLS are based on SSL 3.0, released in 1996.

1997

 PostgreSQL 6.0 was released on January 29, 1997, as the first formal release under the new name, introducing unique indexes and the `pg_dumpall` utility.

 Apple Computer acquired NeXT for \$427 million on February 4, 1997, gaining the OPENSTEP operating system and bringing Steve Jobs back to Apple. The NeXTSTEP-derived codebase would serve as the foundation for Mac OS X.

 Visual Studio is an integrated development environment (IDE) from Microsoft used to develop computer programs, websites, web apps, web services and mobile apps. Microsoft first released Visual Studio on March 19, 1997, bundling many of its programming tools together for the first time as Visual Studio 97.

 The essay "The Cathedral and the Bazaar" by Eric S. Raymond was first presented at the Linux Kongress on May 27, 1997

 The first edition of ECMA-262 (ECMAScript) was adopted by the Ecma General Assembly in June 1997.

 The original version of the IEEE 802.11 standard for wireless local area networks was approved in June 1997, providing data rates of 1 and 2 Mbit/s.

 The unified modeling language (UML) is a general-purpose visual modeling language that is intended to provide a standard way to visualize the design of a system. UML 1.1 was submitted to the OMG in August 1997 and adopted by the OMG in November 1997.

 The Comprehensive R Archive Network (CRAN) is R's central software repository, supported by the R Foundation. CRAN was created by Kurt Hornik and Friedrich Leisch in 1997, with the name paralleling other early packing systems such as TeX's CTAN (released 1992) and Perl's CPAN (released 1995).

 Zeev Suraski and Andi Gutmans rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive acronym PHP: Hypertext Preprocessor.

 A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes. This allows it to exhibit temporal dynamic behavior. Long short-term memory (LSTM) networks were invented by Hochreiter and Schmidhuber in 1997 and set accuracy records in multiple applications domains.

 Google Search is a search engine operated by Google that uses algorithms to analyze and rank websites based on their relevance to the search query, launched in

1997.

 JUnit was created by Kent Beck and Erich Gamma during a flight from Zurich to OOPSLA 1997 in Atlanta, porting Beck's SUnit framework to Java using pair programming and test-first development. It became the dominant testing framework for Java and established automated unit testing as a standard engineering practice.

1998

 Netscape Communications Corporation announced on January 22, 1998, that it would release the source code for its flagship Netscape Communicator product as free software.

 Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data, defining a set of rules for encoding documents in a format that is both human-readable and machine-readable. It was first published on February 10, 1998.

 The Open Source Initiative (OSI) is the steward of the Open Source Definition, the set of rules that define open source software. The organization was founded in late February 1998 by Bruce Perens and Eric S. Raymond, part of a group inspired by Netscape's announcement to open-source its browser suite.

 Netscape Communications Corporation released the source code for Netscape Communicator and started the Mozilla project on March 31, 1998, influenced by "The Cathedral and the Bazaar"

 Advanced package tool, or APT, is a free-software user interface that works with core libraries to handle the installation and removal of software on Debian and Debian-based Linux distributions, with initial release on March 31, 1998.

 Gigabit Ethernet, running at 1000 Mbps, was formally approved as the IEEE 802.3z standard on June 25, 1998.

 The iMac G3 began shipping on August 15, 1998, featuring a translucent Bondi Blue enclosure designed by Jony Ive. It dropped legacy ports like SCSI and floppy drives in favor of USB, and is widely credited with saving Apple from near-bankruptcy after Steve Jobs's return.

 The Document Object Model (DOM) is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. It was first published on October 1, 1998.

🌐 Apache Tomcat is a free and open-source implementation of the Jakarta Servlet, Jakarta Pages, and related Java web technologies. It started off in November 1998 as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems, who chose the name "tomcat" because the animal represented something that could fend for itself.

🌐 Servlet 2.1, released in November 1998, was the first official Servlet specification, adding the RequestDispatcher interface and the ServletContext for sharing information across a web application.

🔒 OpenSSL is a software library for applications that secure communications over computer networks against eavesdropping or need to identify the party at the other end, widely used by Internet servers including the majority of HTTPS websites. The OpenSSL project was founded in 1998 to provide a free set of encryption tools for the code used on the Internet, with initial release on December 23, 1998.

🌐 LAMP (Linux, Apache, MySQL, PHP/Perl/Python) is an acronym denoting one of the most common software stacks for many of the web's most popular applications. The acronym LAMP was coined by Michael Kunze in the December 1998 issue of Computertechnik, a German computing magazine, as he demonstrated that a bundle of free and open-source software "could be a feasible alternative to expensive commercial packages".

⚙️ In December 1998, IPv6 became a Draft Standard for the IETF with the publication of RFC 2460, which superseded RFC 1883.

📖 In 1998, the Ruby Application Archive was launched by Matsumoto, along with a simple English-language homepage for Ruby.

📄 The Halloween documents comprise a series of confidential Microsoft memoranda on potential strategies relating to free software, open-source software, and to Linux in particular, and a series of media responses to these memoranda. Both the leaked documents and the responses were published by open-source software advocate Eric S. Raymond in 1998.

🌐 In 1998, the Microsoft Outlook Web Access team developed the concept behind the XMLHttpRequest scripting object. XMLHttpRequest (XHR) is an API in the form of an object whose methods transfer data between a web browser and a web server. The object is provided by the browser's JavaScript environment.

🔒 AppArmor ("Application Armor") is a Linux kernel security module that allows the system administrator to restrict programs' capabilities with per-program profiles allowing capabilities like network access, raw socket access, and the permission to read, write, or execute files on matching paths. It was initially released in 1998.

🌐 Perl 5 gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its powerful regular expression and string parsing abilities.

⚙️ Between 1998 and 2004, CFEngine grew in adoption along with the popularity of Linux as a computing platform.

1999

🔒 TLS 1.0 was first defined in RFC 2246 in January 1999 as an upgrade of SSL Version 3.0.

📖 The Haskell 98 Report: Language and Libraries was published in February 1999, edited by Simon Peyton Jones and John Hughes. It represented a commitment to stability: a particular instance of the language was named Haskell 98, which implementers committed to supporting indefinitely, while its free-spirited children remained free to term themselves Haskell. The Haskell Committee disbanded in the same year.

⚙️ Mac OS X Server 1.0 was released on March 16, 1999, as the first operating system from Apple built on the NeXTSTEP codebase acquired from NeXT. It used a modified version of the classic Mac OS GUI and served as a precursor to the consumer Mac OS X.

🏢 Salesforce, Inc. is an American cloud-based software company headquartered in San Francisco, California. It provides applications focused on sales, customer service, marketing automation, e-commerce, analytics, artificial intelligence, and application development. Salesforce was founded on March 8, 1999 by former Oracle executive Marc Benioff, together with Parker Harris, Dave Moellenhoff, and Frank Dominguez as a software-as-a-service (SaaS) company.

🏢 The Apache Software Foundation (ASF) is an American nonprofit corporation to support a number of open source software projects. The ASF was formed from a group of developers of the Apache HTTP Server, and incorporated on March 25, 1999.

🦋 The Melissa virus was a fast-spreading macro virus that first appeared around March 26, 1999. The virus mainly attacked computers using Microsoft Word and Outlook.

📖 HotSpot, released as Java HotSpot Performance Engine, is a Java virtual machine for desktop and server computers, developed by Sun Microsystems and now maintained and distributed by Oracle Corporation. It features improved performance via methods such as just-in-time compilation and adaptive optimization. The Java HotSpot Performance Engine was released on April 27, 1999, built on technologies from an implementation of the programming language Smalltalk named Strongtalk. Initially available as an add-on for Java 1.2, HotSpot became the default Sun JVM in Java 1.3.

⚙️ VMware Workstation Pro (known as VMware Workstation until release of VMware Workstation 12 in 2015) is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems, with initial release on May 15, 1999.

⚙️ RRDtool (round-robin database tool) aims to handle time series data such as network bandwidth, temperatures or CPU load, with initial release on July 16, 1999.

🌐 Servlet 2.2, released in August 1999, became part of Java 2 Platform, Enterprise Edition (J2EE) and introduced web application archive (`.war`) files as the standard packaging and deployment format.

🔒 GnuPG was initially developed by Werner Koch, with the first production version (1.0.0) released on September 7, 1999, almost two years after the first GnuPG release (version 0.0.0).

📖 Extreme programming (XP) is a software development methodology intended to improve software quality and responsiveness to changing customer requirements. Kent Beck developed extreme programming during his work. He began to refine the development methodology used in the project and wrote a book on the methodology (Extreme Programming Explained, published in October 1999).

📖 The Pragmatic Programmer: From Journeyman to Master is a book about computer programming and software engineering, written by Andrew Hunt and David Thomas and published in October 1999.

💻 GNU nano is a text editor for Unix-like computing systems or operating environments using a command line interface, with initial release on November 18, 1999.

🔒 OpenBSD 2.6 was released on December 1, 1999, featuring the first release of OpenSSH. Originally developed as a free replacement for the proprietary SSH suite, OpenSSH is based on the Secure Shell (SSH) protocol and provides a secure channel over an unsecured network in a client-server architecture.

📖 Jakarta EE, formerly Java Platform, Enterprise Edition (Java EE) and Java 2 Platform, Enterprise Edition (J2EE), is a set of specifications extending Java SE with specifications for enterprise features such as distributed computing and web services, with initial specification release on December 17, 1999.

🧑‍🤝‍🧑 *Peopleware: Productive Projects and Teams* was revised for its 2nd Edition in 1999

📖 The book *The Cathedral and the Bazaar* was published in 1999 and was released under the Open Publication License v2.0 in the same year.

📖 Nikolai Bezroukov published two critical essays on Eric Raymond's views of open

source software in 1999

 Apache Lucene is a free and open-source search engine software library, originally written in 1999 by Doug Cutting. The name Lucene is Doug Cutting's wife's middle name and her maternal grandmother's first name.

 In 1999, James Duncan Davidson helped make Tomcat open-source and played a key role in its donation by Sun Microsystems to the Apache Software Foundation, where it became part of the Jakarta Project. Tomcat 3.0.x, the first Apache release, implemented the Servlet 2.2 and JSP 1.1 specifications.

 JavaServer Pages (JSP) is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but uses the Java programming language.

 SourceForge, founded in 1999 by VA Software, was the first provider of a centralized location for free and open-source software developers to control and manage software development and offering this service without charge.

Timeline - 2000-09

2000

 Flash 5, released in 2000, introduced ActionScript 1.0 — a full scripting language based on ECMAScript — enabling complex interactivity and turning Flash into a platform for rich internet applications, games, and video players.

 The jail mechanism is an implementation of FreeBSD's OS-level virtualisation that allows system administrators to partition a FreeBSD-derived computer system into several independent mini-systems called jails, first introduced in FreeBSD version 4.0 on March 14, 2000. Jails aim at three primary goals: virtualization, security, and ease of delegation.

 In March 2000, the Nupedia project was started by Jimmy Wales and Larry Sanger as a free, peer-reviewed encyclopedia — the predecessor project whose slow growth led directly to Wikipedia's creation the following year.

 On 22 May 2000, PHP 4, powered by the Zend Engine 1.0, was released.

 SQLite is a database engine written in the C programming language, first released on August 17, 2000. It is not a standalone app but a library that software developers embed in their apps, making it the most widely deployed database engine in the world.

 Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

 Apache Subversion (SVN) is a software versioning and revision control system. CollabNet founded the project in 2000 to write an open-source version-control system that operated like CVS but fixed its bugs and added missing features, with its first release on October 20, 2000.

 Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies, including mandatory access controls (MAC). It was first released on December 22, 2000.

 C# (pronounced see sharp) is a general-purpose, multi-paradigm programming language designed by Anders Hejlsberg at Microsoft in 2000. It was later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO/IEC (ISO/IEC 23270) in 2003.

 CMake is a cross-platform build system generator developed in response to the need for a unified build environment for the Insight Segmentation and Registration Toolkit (ITK), first implemented in 2000 and further developed in 2001.

 The Linux Foundation (LF) is a non-profit technology consortium founded in 2000 as a merger between Open Source Development Labs and the Free Standards Group to standardize Linux, support its growth, and promote its commercial adoption.

 In 2000, Roy Fielding proposed Representational State Transfer (REST) as an architectural approach to designing web services. REST is an architectural style for building distributed systems based on hypermedia.

 Microsoft officially announced the .NET initiative at the Forum 2000 conference in June 2000, presenting a new "managed" software platform intended to address the complexities of binary-level component models like COM.

2001

 Linux version 2.4.0, released on 4 January 2001, contained support for ISA Plug and Play, USB, and PC Cards. Linux 2.4 added support for the Pentium 4 and Itanium, and for the newer 64-bit MIPS processor.

 Creative Commons (CC) is an American non-profit organization and international network devoted to educational access and expanding the range of creative works available for others to build upon legally and to share, founded on January 15, 2001.

 Wikipedia is a free online encyclopedia created and edited by volunteers around the

world. It was founded on January 15, 2001, by Jimmy Wales and Larry Sanger, based on the principle of creating a comprehensive, accessible, and reliable source of information that anyone could edit. Wikipedia runs on MediaWiki, software created specifically for Wikipedia's needs.

🏢 On February 11-13, 2001, seventeen software industry leaders met to talk, ski, relax, and try to find common ground, and together published the Manifesto for Agile Software Development, which counts Scrum co-creators Ken Schwaber and Jeff Sutherland among its signatories.

⚙️ VMware ESXi (formerly ESX), an enterprise-class type-1 hypervisor developed by VMware for deploying and serving virtual computers, was first released on March 23, 2001.

⚙️ Mac OS X 10.0 Cheetah was released on March 24, 2001, as the first desktop version of Apple's Unix-based operating system built on the NeXTSTEP foundation. It introduced the Aqua user interface, replacing the classic Mac OS look and feel.

☁️ CruiseControl, a Java-based continuous build framework distributed under a BSD-style license, was first released on March 30, 2001 as one of the earliest open-source continuous integration tools.

🗄️ PostgreSQL 7.1, released on April 13, 2001, introduced the Write-Ahead Log (WAL), a critical feature for ensuring data durability and crash recovery.

📄 YAML is a human-readable data-serialization language commonly used for configuration files and data storage or transmission. It was first released on May 11, 2001.

📄 reStructuredText (RST) is a file format for textual data used primarily in the Python community for technical documentation. Part of the Docutils project, it was first released on June 1, 2001 to provide Python with documentation tools similar to Javadoc for Java.

🌐 WebKit is a browser engine developed by Apple and primarily used in its Safari web browser and all iOS web browsers, started by Don Melton within Apple on June 25, 2001 as a fork of KHTML and KJS.

🐛 Code Red was a computer worm observed on the Internet on July 15, 2001 that attacked computers running Microsoft's IIS web server, becoming the first large-scale mixed-threat attack to successfully target enterprise networks.

🐛 The Nimda virus is a malicious file-infecting computer worm that quickly spread, surpassing the economic damage caused by previous outbreaks such as Code Red, with the first released advisory appearing on September 18, 2001.

 In September 2001, Lucene joined the Apache Software Foundation's Jakarta family of open-source Java products.

 Tomcat 4.0, released in September 2001, was a complete rewrite implementing Servlet 2.3 and JSP 1.2, and introduced its enduring architecture of Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

 Windows XP, launched on October 25, 2001, finally merged Microsoft's consumer and business operating systems by moving the consumer line onto the stable NT kernel, featuring a redesigned "Luna" interface that would dominate the desktop market for over a decade.

 Eclipse is an integrated development environment (IDE) with a base workspace and an extensible plug-in system, inspired by the Smalltalk-based VisualAge family of IDEs. Eclipse 1.0 was released on November 29, 2001, and was the most popular Java IDE until 2016.

 SciPy is a free and open-source Python library for scientific and technical computing created around 2001 when Travis Oliphant, Eric Jones, and Pearu Peterson merged code they had written and unified it under a single package name.

 VMware Server (formerly VMware GSX Server) is a discontinued free-of-charge virtualization software server suite developed by VMware, Inc., with both ESX 1.0 (Type 1 hypervisor) and GSX 1.0 (Type 2 hypervisor) launching in 2001.

 IPython (Interactive Python) is a command shell for interactive computing originally developed for Python, offering introspection, rich media, shell syntax, tab completion, and history. It was first released in 2001.

 SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA) and first published in 2001, built using the Merkle–Damgård construction from a one-way compression function itself based on the Davies–Meyer structure from a specialized block cipher.

 The Advanced Encryption Standard (AES), also known by its original name Rijndael, is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

 In 2001, the original Java Specification Request (JSR) proposed developing a package with the name `javax.servlet.ui`, and in June 2001, JavaWorld reported on Amy Fowler's team's design of "the JavaServer Faces API" (also known as "Moonwalk") as "an application framework for creating Web-based user interfaces"

 Toyota formalized its internal management values as "The Toyota Way" in 2001, consisting of two main pillars: Continuous Improvement (Kaizen) and Respect for

People.

2002

🌐 ASP.NET is a server-side web-application framework for building dynamic web pages, first released on January 5, 2002 as part of the .NET Framework 1.0. It is the successor to Microsoft's Active Server Pages (ASP) technology.

📦 .NET Framework 1.0 was officially released on February 13, 2002, alongside Visual Studio .NET, introducing the Common Language Runtime (CLR) and the C# programming language.

⚙️ Arch Linux is an independently developed, x86-64 general-purpose Linux distribution that strives to provide the latest stable versions of most software by following a rolling-release model. It was started by Judd Vinet and first released on March 11, 2002.

⚙️ Gentoo Linux is a Linux distribution built using the Portage package management system, with version 1.0 released on March 31, 2002, and the non-profit Gentoo Foundation established in 2004 with all copyrights and trademarks transferred to it.

🌐 The Mozilla project developed and implemented an interface called nsXMLHttpRequest into the Gecko layout engine, creating a wrapper that exposed it as the XMLHttpRequest JavaScript object. The XMLHttpRequest object was accessible as early as Gecko version 0.6 (released December 6, 2000) but was not completely functional until Gecko 1.0 (released June 5, 2002).

🌐 Mozilla Firefox is a free and open-source web browser developed by the Mozilla Foundation, using the Gecko rendering engine to display web pages. It was first released on September 23, 2002.

📖 The Spring Framework is an application framework and inversion of control container for the Java platform, first released on October 1, 2002. Its core features can be used by any Java application, with extensions for building web applications on top of Java EE.

🧠 Torch is an open-source machine learning library and scientific computing framework based on the Lua programming language, first released in October 2002.

📖 "Test-Driven Development: By Example" by Kent Beck was published by Addison-Wesley in November 2002, popularizing the red-green-refactor cycle and presenting test-driven development as a discipline in which automated tests are written before the production code that makes them pass.

⚙️ The Linux Namespaces originated in 2002 in the 2.4.19 kernel with work on the mount namespace kind. Additional namespaces were added beginning in 2006 and continuing into the future.

📄 AsciiDoc is a human-readable document format semantically equivalent to DocBook XML but using plain-text markup conventions, created in 2002 by Stuart Rackham along with Python-based tools ('asciidoc' and 'a2x') for converting plain-text files to commonly used published document formats.

🌐 JSON is a language-independent data format derived from JavaScript, with the JSON.org website launched in 2002 to establish it as a standard data interchange format.

📊 The final major release of the software, Lotus 1-2-3 Release 9.8, was launched in 2002 as part of the Lotus SmartSuite office package.

🏢 Ken Schwaber founded the Scrum Alliance in 2002 to provide worldwide Scrum training and certification.

📖 Kent Beck published Test Driven Development: By Example on November 8, 2002, formalising TDD as a standalone practice with the red-green-refactor cycle: write a failing test, make it pass with minimal code, then refactor.

2003

📊 Tableau Software, LLC is an American interactive data visualization software company focused on business intelligence, formally founded in January 2003 by Pat Hanrahan, Christian Chabot, and Chris Stolte, with its headquarters moving to Seattle, Washington in 2004.

🌐 The Wikimedia Foundation, the non-profit organization that has hosted Wikipedia ever since, was incorporated in St. Petersburg, Florida, on June 20, 2003.

💻 The first public demonstration of Monad (Microsoft Shell or MSH) occurred in September 2003 at the Professional Developers Conference (PDC).

⚙️ Android, Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White to develop "smarter mobile devices that are more aware of its owner's location and preferences"

⚙️ Linux Version 2.6.0 was released on 17 December 2003. The development for 2.6.x changed further towards including new features throughout the duration of the series.

📄 The Python Package Index (PyPI), also known as the Cheese Shop, is the official third-party software repository for Python, launched in 2003. Its metadata standard

was defined by PEP 241 in March 2001, with the comprehensive catalog proposal finalized in November 2002.

 Domain-driven design (DDD) is a major software design approach focusing on modeling software to match a domain according to input from domain experts, with the term coined by Eric Evans in his book of the same name published in 2003.

 Google Borg is a cluster manager used by Google. It led to widespread use of similar approaches such as Docker and Kubernetes. According to the research paper published by Google in 2015, Borg was developed in 2003.

 Xen is a type-1 hypervisor that provides services allowing multiple computer operating systems to execute on the same hardware concurrently, originating as a research project at the University of Cambridge led by Ian Pratt and his PhD student Keir Fraser. The first public release was made in 2003, with v1.0 following in 2004, and the project is now developed by the Linux Foundation.

 Matplotlib is a plotting library for the Python programming language and NumPy, providing an object-oriented API for embedding plots into applications using GUI toolkits like Tkinter, wxPython, Qt, or GTK. It was first released in 2003.

 A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a challenge-response Turing test used in computing to determine whether a user is human and to deter bot attacks and spam, with the term coined in 2003 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford.

 RFC 3629 was published in November 2003, updating the UTF-8 specification to restrict it to the range U+0000..U+10FFFF and establishing UTF-8 as a standard Internet protocol element.

 Dan North introduced behaviour-driven development (BDD) in late 2003 while writing JBehave, a replacement for JUnit that removed testing vocabulary in favour of one centred on verifying behaviour. Together with Chris Matts he extended the approach to requirements via the "Given/When/Then" scenario vocabulary.

 In 2003, a poll conducted by GameDev.net showed that Lua was the most popular scripting language for game programming.

2004

 Scala is a strong statically typed general-purpose programming language supporting both object-oriented and functional programming paradigms, first released on January 20, 2004.

🌐 JavaServer Faces 1.0 was released on March 11, 2004, as the initial specification for a component-based web application framework

📖 RubyGems is a package manager for the Ruby programming language that provides a standard format for distributing Ruby programs and libraries, with development starting in November 2003 and public release on March 14, 2004 (Pi Day).

🌐 Google launched Gmail in a limited beta release on April 1, 2004, offering one gigabyte of storage—significantly more than competitors at the time

📖 In April 2004, Windows Installer XML (WiX) was the first Microsoft project to be released under an open-source license, the Common Public License. Initially hosted on SourceForge, it was also the first Microsoft project to be hosted externally.

⚙️ In April 2004, South African entrepreneur Mark Shuttleworth invited a dozen Debian developers to his London flat, where they brainstormed and laid out the distinguishing features of what would become Ubuntu. To fund the project, Shuttleworth created Canonical Ltd. to employ the developers using his fortune from selling Thawte to Verisign.

🌐 The Web Hypertext Application Technology Working Group (WHATWG) was announced on June 4, 2004, founded by Apple, Mozilla, and Opera after the W3C rejected their joint proposal for evolving HTML at a workshop. WHATWG began independently developing a new HTML specification that would eventually become HTML5.

🌐 On 1 July 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension, and numerous performance enhancements.

📖 Maven is a build automation tool used primarily for Java projects. Created by Jason van Zyl, it began as a sub-project of Apache Turbine in 2002 and was accepted as a top-level Apache Software Foundation project in 2003. Maven 1.0 was released on July 13, 2004.

🌐 Ruby on Rails 0.5.0, the initial open-source release extracted from the project management tool Basecamp, was first released by David Heinemeier Hansson in July 2004.

🌐 Nginx is a web server that can also be used as a reverse proxy, load balancer, and HTTP cache. Created by Igor Sysoev to solve the C10k problem, it was first publicly released on October 4, 2004 and was serving 500 million requests per day for the Rambler portal by September 2008.

⚙️ Ubuntu 4.10 (Warty Warthog) was released on 20 October 2004 as the inaugural

release of Ubuntu. It was built upon Debian, with plans for a new release every six months and eighteen months of support thereafter.

⚙️ Unionfs is a filesystem service for Linux, FreeBSD and NetBSD which implements a union mount for other file systems, allowing files and directories of separate file systems to be transparently overlaid into a single coherent file system. Version 1.0.2 was released on November 9, 2004.

🌐 Version 1.0 of Firefox was released on November 9, 2004. This was followed by version 1.5 in November 2005, version 2.0 in October 2006, version 3.0 in June 2008, version 3.5 in June 2009.

📄 Markdown is a lightweight markup language for creating formatted text using a plain-text editor. John Gruber and Aaron Swartz created Markdown in 2004 as a markup language that is intended to be easy to read in its source code form.

📊 MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster, introduced by Google in 2004.

🧠 In 2004, K. S. Oh and K. Jung demonstrated that standard neural networks can be greatly accelerated on GPUs, achieving implementations 20 times faster than CPU equivalents, with subsequent papers in 2005 further emphasizing the value of GPGPU for machine learning.

🌐 Selenium was created in 2004 by Jason Huggins at ThoughtWorks in Chicago as "JavaScriptTestRunner" to automate testing of an internal time-and-expenses application. Fellow ThoughtWorker Paul Hammant proposed open-sourcing it and co-developed Selenium Remote Control (RC), enabling browser automation from any language over the wire.

2005

📊 PostgreSQL 8.0 was released on January 19, 2005, notable for being the first version with native Microsoft Windows support, as well as adding savepoints and point-in-time recovery.

☁️ Hudson, a continuous integration server written in Java, was created by Kohsuke Kawaguchi at Sun Microsystems and first released on February 7, 2005. After Oracle acquired Sun in 2010, a trademark dispute led the community to fork the project and rename it Jenkins in 2011.

💻 fish (an acronym for "friendly interactive shell") is a Unix shell with a focus on interactivity and usability, first released by Axel Liljencrantz on February 13, 2005.

Unlike most other shells, it does not adhere to POSIX shell standards by design, instead prioritizing user experience with features such as syntax highlighting and autosuggestions.

🌐 The Prototype JavaScript Framework is a JavaScript framework created by Sam Stephenson in February 2005 as part of Ajax support in Ruby on Rails.

📊 In February 2005, Lucene became its own top-level Apache project, no longer a subproject of Jakarta.

🌐 On April 1, 2005, Gmail's first anniversary, Google doubled its storage space to 2 GB.

🔒 Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorization data between parties, particularly between an identity provider and a service provider, with SAML 2.0 becoming an OASIS Standard in March 2005.

⚙️ At WWDC on June 6, 2005, Steve Jobs announced Apple's plan to transition the entire Macintosh line from PowerPC to Intel processors, marking the second major processor architecture transition in Mac history after the shift from Motorola 68000 to PowerPC in 1994.

⚙️ `collectd` is a Unix daemon that collects, transfers and stores performance data of computers and network equipment, first released on July 8, 2005.

⚙️ Google acquired Android, Inc. on July 11, 2005, making it a wholly owned subsidiary of Google and retaining the company's key employees to continue development of the mobile operating system

🌐 Django is a free and open-source, Python-based web framework that follows the model–template–views (MTV) architectural pattern, first released on July 21, 2005.

📖 RSpec was started in August 2005 by Steven Baker with Dave Astels, Aslak Hellesøy, and David Chelimsky as a behaviour-driven development (BDD) framework for Ruby. Inspired by JBehave, it introduced the `describe/it` DSL for expressing software behaviour as executable examples. RSpec 1.0 was released on May 18, 2007, after David Chelimsky took over project leadership in 2006.

📊 In December 2005, Yahoo! began offering some of its Web services in JSON.

🌐 JSONP, or JSON-P (JSON with Padding), is a historical JavaScript technique for requesting data by loading a `<script>` element, which is an element intended to load ordinary JavaScript. JSONP enables sharing of data bypassing same-origin policy. The original proposal for JSONP, where the padding is a callback function, appears to have been made by Bob Ippolito in December 2005.

🌐 Ruby on Rails 1.0 was released on December 13, 2005, marking the framework's first major stable version.

📄 F# (pronounced F sharp) is a functional-first, general-purpose, strongly typed, multi-paradigm programming language developed by the F# Software Foundation, Microsoft, and open contributors, first released in 2005.

💻 Git was created by Linus Torvalds in 2005 for Linux kernel development with contributions from other kernel developers, with maintenance transferred to Junio Hamano on 26 July 2005. Hamano released version 1.0 on 21 December 2005 and has remained the project's core maintainer.

☁ Puppet is a configuration management tool founded by Luke Kanies in 2005, using a declarative language to manage the IT infrastructure lifecycle including provisioning, patching, and configuration. Puppet itself is written in Ruby, first released in 2005.

🏢 Adobe Systems acquired Macromedia on December 3, 2005, for approximately \$3.4 billion, taking ownership of Flash, Dreamweaver, Director, Fireworks, and Authorware. The product was subsequently rebranded as Adobe Flash.

🔗 AOL discontinued Usenet access in 2005, citing the growing popularity of weblogs, chat forums, and online conferencing.

2006

⚙ On January 10, 2006, Steve Jobs introduced the first Intel-based Macs — the iMac and MacBook Pro — at Macworld Conference & Expo, beginning a transition from PowerPC that would be completed across the entire product line within nine months.

📊 SQLAlchemy is an open-source SQL toolkit and object-relational mapper for the Python programming language. It was first released in February 2006.

☁ Amazon S3 (Amazon Simple Storage Service) is an object storage service offered by Amazon Web Services (AWS) through a web service interface, launched in the United States on March 14, 2006.

📊 Apache Hadoop, a framework for distributed processing of large data sets across clusters of computers, was created by Doug Cutting and Mike Cafarella and first released on April 1, 2006. Inspired by Google's MapReduce and Google File System papers, its core consists of the Hadoop Distributed File System (HDFS) and a MapReduce processing model. Cutting named it after his son's toy stuffed elephant.

📄 On 5 April 2006, the World Wide Web Consortium (W3C) released the first draft specification for the XMLHttpRequest object in an attempt to create an official Web

standard.

 Microsoft officially renamed Monad to Windows PowerShell on April 25, 2006, and released the first Release Candidate (RC1) for the tool.

 Java EE 5, released on May 11, 2006, changed the platform's name from J2EE to Java EE and greatly simplified development by adopting Java annotations and generics, along with EJB 3.0 and JPA.

 Upstart was an event-based replacement for the traditional init daemon, first released on August 24, 2006, used by several Unix-like operating systems to perform tasks when the computer is started.

 Amazon Elastic Compute Cloud (EC2) is a cloud computing service within Amazon Web Services (AWS) that allows users to rent virtual computers for running their applications, announced in limited public beta on August 25, 2006 and initially using Xen virtualization exclusively.

 Apple announced on August 7, 2006, that it would ship Ruby on Rails with Mac OS X v10.5 "Leopard," which was later released in October 2007 as the first major operating system to pre-install the framework.

 jQuery is a JavaScript library designed to simplify HTML DOM traversal and manipulation, event handling, CSS animation, and Ajax. Created by John Resig at BarCamp NYC in January 2006, inspired by Dean Edwards' cssQuery library, it was first released on August 26, 2006.

 Windows PowerShell 1.0 was released on November 14, 2006, for Windows XP SP2, Windows Server 2003, and Windows Vista.

 NumPy is a library for the Python programming language that adds support for large multi-dimensional arrays and matrices along with high-level mathematical functions, created in 2006 when Travis Oliphant unified the community around a single array package by porting Numarray's features to Numeric as part of the SciPy project.

 In 2006, Geoffrey Hinton developed the deep belief network technique for training many-layered deep autoencoders.

 In 2006, a revised version of the SSH protocol, SSH-2, was adopted as a standard, introducing incompatibilities with the original SSH-1 protocol.

 Dan North published "Introducing BDD" in 2006, coining the term behaviour-driven development and formalising the Given/When/Then scenario structure as a bridge between business stakeholders and developers.

2007

🌐 On April 10, 2007, Mozilla, Apple, and Opera proposed that the W3C's new HTML Working Group adopt the WHATWG's HTML5 draft as its starting point, formally uniting the two parallel standardization efforts under the HTML5 name.

⚙️ Steve Jobs unveiled the first-generation iPhone, running what was then marketed as a version of "OS X," during his keynote address at the Macworld Conference & Expo on January 9, 2007.

📄 Apache Groovy is a Java-syntax-compatible, statically and dynamically typed object-oriented programming language for the Java platform with features similar to Python, Ruby, and Smalltalk, first discussed by James Strachan in August 2003 and formally released as version 1.0 on January 2, 2007 following Java Community Process standardization efforts.

🌐 Simon Stewart pushed the initial commit of WebDriver on January 3, 2007, developing a browser automation tool that controlled browsers natively rather than through injected JavaScript, overcoming the security sandbox limitations of Selenium RC.

⚙️ Oracle VM VirtualBox (formerly Sun VirtualBox and Innotek VirtualBox) is a type-2 hypervisor for x86 virtualization developed by Oracle Corporation, first released on January 17, 2007.

📊 Excel 2007, released on January 30, 2007, introduced the Ribbon interface and the new XML-based file format .xlsx, replacing the binary .xls format used in previous versions.

⚙️ Kernel-based Virtual Machine (KVM) is a virtualization module in the Linux kernel that allows the kernel to function as a hypervisor. It was merged into the mainline Linux kernel in version 2.6.20, which was released on February 5, 2007.

📄 Sun released the Java HotSpot virtual machine and compiler as free software under the GNU General Public License on November 13, 2006, with a promise that the rest of the JDK (which includes the Java Runtime Environment) would be placed under the GPL by March 2007.

📄 Rake is a Make-like build automation program implemented in Ruby, where tasks and dependencies are specified in standard Ruby syntax. Version 0.7.3 was released on April 21, 2007.

📄 OpenJDK (Open Java Development Kit) is a free and open-source implementation of Java SE, the result of an effort Sun Microsystems began in 2006. It produces the

HotSpot virtual machine, the Java Class Library, and the javac compiler, with its first release on May 8, 2007.

🧠 The scikit-learn project started as scikits.learn, a Google Summer of Code project by French data scientist David Cournapeau, first released in June 2007.

⚙️ iPhone OS 1, the first version of Apple's mobile operating system, was released alongside the original iPhone on June 29, 2007.

☁️ Heroku was founded in June 2007 by James Lindenbaum, Adam Wiggins, and Orion Henry as a cloud platform-as-a-service (PaaS) initially supporting Ruby web applications, with the prototype taking around six months to develop. The platform received \$3 million in funding in May 2008.

📖 PyPy is an alternative implementation of the Python programming language to CPython (the standard implementation) that often runs faster due to its just-in-time compiler, starting as a research project and reaching a mature 1.0 release in mid-2007 with a subsequent focus on production-readiness and enhanced CPython compatibility.

🌐 Sinatra was created and open-sourced by Blake Mizerany on September 9, 2007, as a lightweight Domain Specific Language (DSL) for building web applications in Ruby.

🌐 In October 2007, Gmail added IMAP support and increased storage to 4 GB, following competitors' changes.

🌐 Development of the GitHub.com platform began on October 19, 2007. The site was launched in April 2008 by Tom Preston-Werner, Chris Wanstrath, P. J. Hyett and Scott Chacon after it had been made available for a few months prior as a beta release.

📖 Language Integrated Query (LINQ) is a Microsoft .NET Framework component that adds native data querying capabilities to .NET languages, originally released as a major part of .NET Framework 3.5 on 19 November 2007.

📖 The C# language v3.0, released in November 2007 with .NET Framework v3.5, also has full support of anonymous functions.

📖 F# added asynchronous workflows with await points in version 2.0 in 2007. This influenced the async/await mechanism added to C#.

📖 Go was conceived in the second half of 2007 at Google by Robert Griesemer, Rob Pike, and Ken Thompson, motivated by dissatisfaction with the complexity and slow compilation times of C++.

2008

 Pandas is a Python library for data manipulation and analysis, offering data structures and operations for manipulating numerical tables and time series. It was first released on January 11, 2008.

 HTML5 is a markup language used for structuring and presenting content on the World Wide Web. The first public W3C draft was released on January 22, 2008, introducing landmark features including `<canvas>`, `<video>`, `<audio>`, local storage, geolocation, Web Workers, and WebSockets — collectively enabling rich web applications without browser plugins.

 The control groups functionality was merged into the Linux kernel mainline in kernel version 2.6.24, which was released in January 2008.

 FreeBSD 7.0-RELEASE was released on February 27, 2008. This version added experimental support for the ZFS file system and introduced the ULE scheduler to improve performance on multi-core systems.

 Sphinx is a documentation generator written in Python and widely used by the Python community, first released on March 21, 2008. It converts reStructuredText files into HTML websites and other formats including PDF, EPub, Texinfo, and man pages.

 HBase is an open-source non-relational distributed database modeled after Google's Bigtable and written in Java, first released on March 28, 2008. It runs on top of HDFS or Alluxio as part of the Apache Hadoop project, providing Bigtable-like capabilities for Hadoop.

 Gradle is a build automation tool for multi-language software development, covering the full lifecycle from compilation and packaging to testing, deployment, and publishing. It was first released on April 21, 2008.

 On April 7, 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company.

 PDF was published as an open ISO standard on July 1, 2008, as ISO 32000-1:2008, ending Adobe's proprietary control over the format. The standardized version was based on PDF 1.7 and was approved by ISO Technical Committee 171.

 iPhone OS 2 was released on July 11, 2008, introducing the App Store and support for third-party applications, which significantly expanded the platform's capabilities.

 Jinja is a web template engine for the Python programming language. It was created by Armin Ronacher and is licensed under a BSD License, with its initial release

on July 17, 2008.

 Apache Cassandra is a free and open-source, distributed, wide-column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It was initially released in July 2008.

 Linux Containers (LXC), an operating-system-level virtualization method for running multiple isolated Linux systems on a single kernel, was first released on August 6, 2008. It combines the kernel's cgroups and namespace isolation to provide lightweight containers, and served as the foundation upon which Docker was originally built before Docker replaced it with its own libcontainer in 2014.

 The domain name bitcoin.org was registered on 18 August 2008.

 On August 20, 2008, Amazon added Elastic Block Store (EBS) to provide persistent storage, a critical feature that had been lacking since EC2's introduction.

 TLS 1.2 was defined in RFC 5246 in August 2008.

 Google Chrome is a cross-platform web browser developed by Google. It was first released on September 2, 2008, for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox.

 V8 is the JavaScript execution engine which was initially built for Google Chrome. It was then open-sourced by Google in 2008, with its first version released at the same time as the first version of Chrome on September 2, 2008. Much of V8's development is strongly inspired by the Java HotSpot Virtual Machine developed by Sun Microsystems, with the newer execution pipelines being very similar to those of HotSpot's.

 Stack Overflow is a question and answer website for computer programmers. It was launched on September 15, 2008, by Jeff Atwood, Joel Spolsky, and Jarrod Dixon. Stack Overflow has become the largest online community for programmers and serves as a central repository of programming knowledge, allowing users to ask questions, provide answers, and vote on the quality of contributions.

 Android 1.0, the first commercial version of the operating system, was released on September 23, 2008, featuring an integration with Google services and the introduction of the Android Market.

 DuckDuckGo is an American software company focused on online privacy whose flagship product is a search engine named DuckDuckGo. It was launched on September 25, 2008.

 Open Virtualization Format (OVF) is an open standard for packaging and

distributing virtual appliances or, more generally, software to be run in virtual machines. It was initially released in September 2008.

✂ On 31 October 2008, a link to a white paper authored by Satoshi Nakamoto titled Bitcoin: A Peer-to-Peer Electronic Cash System was posted to a cryptography mailing list.

📖 Python 3.0 was released on 3 December 2008 as a major revision of the language that intentionally breaks backward compatibility to improve clarity and remove legacy complexity.

🏢 In 2008, Microsoft joined the Apache Software Foundation and co-founded the Open Web Foundation with Google, Facebook, Sun, IBM, Apache, and others.

⚙ Graphite is a free open-source software (FOSS) tool that monitors and graphs numeric time-series data such as the performance of computer systems. Graphite was developed by Orbitz Worldwide, Inc and released as open-source software in 2008.

✂ UTF-8 became the most common character encoding on the World Wide Web in 2008, surpassing ASCII and other legacy encodings.

📖 Cucumber was created in 2008 by Aslak Hellesøy as a replacement for the RSpec Story Runner (formerly RBehave), enabling plain-language Gherkin scenarios to be shared between developers and non-technical stakeholders. The name was suggested by Hellesøy's fiancée after he initially called the project "Stories."

2009

☁ In January 2009, Heroku launched a new version of the platform after a three-month rebuild.

☁ Progress Chef (formerly Chef) is a configuration management tool written in Ruby and Erlang. It uses a pure-Ruby, domain-specific language (DSL) for writing system configuration "recipes". It was initially released in January 2009.

🌐 Cross-Origin Resource Sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served, with the draft specification renamed to its final form in March 2009.

⚙ Android 1.5 Cupcake was released on April 27, 2009, initiating the tradition of naming major versions after confectionery items and introducing key features like the on-screen keyboard and support for third-party widgets.

🏢 WolframAlpha is an answer engine developed by Wolfram Research. The engine is

based on Wolfram's earlier product Wolfram Mathematica, a technical computing platform. It was launched on May 18, 2009.

⚙️ Homebrew is a free and open-source software package management system that simplifies the installation of software on Apple's operating system, macOS, as well as Linux. Originally written by Max Howell, the package manager has gained popularity in the Ruby on Rails community and earned praise for its extensibility. It was initially released on May 21, 2009.

🌐 JavaServer Faces 2.0 was released on July 1, 2009, as a significant release for ease of use, enhanced functionality, and performance, coinciding with Java EE 6, where Facelets was adopted as the official view technology, replacing JSP

🌐 Gmail exited beta status on July 7, 2009.

💻 Windows PowerShell 2.0 was released in July 2009 as an integral part of Windows 7 and Windows Server 2008 R2, introducing key features such as PowerShell Remoting and background jobs.

💻 Oh My Zsh, a community-driven framework for managing Zsh configuration, was born on August 28, 2009, when Robby Russell packaged his shell configuration into a Git repository and put it on GitHub so the coworkers he collaborated with every day could improve it with him. By 2024 it had grown to over 2,300 contributors, over 300 plug-ins, and over 140 themes.

📖 CommonJS is a project with the goal to establish conventions on the module ecosystem for JavaScript outside of the web browser. The project was started by Mozilla engineer Kevin Dangoor in January 2009 and initially named ServerJS. In August 2009, the project was renamed CommonJS to show the broader applicability of the APIs.

💻 In September 2009, Joel Spolsky's company Fog Creek Software released a beta version of the Stack Exchange 1.0 platform, extending the Stack Overflow question-and-answer model into a broader network of sites.

📊 Amazon Relational Database Service (or Amazon RDS) is a distributed relational database service by Amazon Web Services (AWS). It was first released on October 22, 2009, supporting MySQL databases, followed by support for Oracle Database in June 2011, Microsoft SQL Server in May 2012, and PostgreSQL in November 2013.

⚙️ Following the ambitious but often criticized Windows Vista (2007), Windows 7 was released on October 22, 2009, refining the user interface and significantly improving performance, leading to its widespread adoption in both corporate and home environments.

⚙️ VMware Server reached its final release on October 26, 2009, after which VMware

discontinued the product in favor of its commercial ESXi hypervisor.

 MariaDB is a community-developed, commercially supported fork of the MySQL relational database management system (RDBMS), intended to remain free and open-source software under the GNU General Public License. It was initially released on October 29, 2009.

 The IEEE 802.11n amendment (later branded Wi-Fi 4) was approved and published in October 2009.

 Node.js was written initially by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web. Dahl demonstrated the project at the inaugural European JSConf on November 8, 2009.

 Go is a statically typed, compiled programming language designed at Google by Robert Griesemer, Rob Pike, and Ken Thompson. It was publicly announced on November 10, 2009, as an open-source project.

 Haskell 2010, the first revision of the Haskell 98 standard, was announced in November 2009 and published in July 2010. It added several well-used and uncontroversial features previously enabled via compiler-specific flags, including the foreign function interface (FFI), hierarchical module names, and pattern guards, while so-called n+k patterns were no longer allowed.

 DevOps as a term originated in 2009 following a talk at the O'Reilly Velocity Conference titled "10+ Deploys per Day: Dev and Ops Cooperation at Flickr." John Allspaw and Paul Hammond walked through some of the pains in the current software development lifecycle.

 In 2009, the first conference named devopsdays was held in Ghent, Belgium. The conference was founded by Belgian consultant, project manager and agile practitioner Patrick Debois.

 Microsoft first began contributing to the Linux kernel in 2009.

 Google Chrome 4 was the first browser to ship full WebSocket support enabled by default in December 2009, proving real-world feasibility of persistent bidirectional browser-server communication.

 SPDY is an obsolete open-specification communication protocol developed for transporting web content. Google announced SPDY in late 2009 and deployed it in 2010.

 Nakamoto implemented the bitcoin software as open-source code and released it in

January 2009.

Timeline - 2010-14

2010

 npm is a package manager for the JavaScript programming language written entirely in JavaScript and developed by Isaac Z. Schlueter to address problems he had "seen with module packaging done terribly," drawing inspiration from similar projects like PEAR (PHP) and CPAN (Perl), first released on January 12, 2010.

 Steve Jobs announced the original iPad on January 27, 2010, at the Yerba Buena Center for the Arts in San Francisco, describing it as a new category of device between a smartphone and a laptop. The Wi-Fi model was released on April 3, 2010, running iPhone OS 3.2.

 Windows Azure Platform became commercially available on February 1, 2010.

 Elasticsearch is a distributed, RESTful search and analytics engine built on Apache Lucene. Shay Banon created its precursor, Compass, in 2004, and released the first version of Elasticsearch on February 8, 2010, describing it as "You Know, for Search."

 Ken Schwaber founded Scrum.org in 2009, and the first official version of "The Scrum Guide" was published by him and Jeff Sutherland in February 2010.

 systemd is a software suite providing an array of system components for Linux operating systems designed to unify service configuration and behavior across distributions. Developed by Red Hat software engineers Lennart Poettering and Kay Sievers as a project to replace Linux's conventional System V init, systemd was first released on March 30, 2010.

 Sinatra 1.0 was released on March 23, 2010, marking the framework's first major stable version and introducing **Rack::Test** as the recommended testing tool.

 Vagrant is an open-source software product for building and maintaining portable virtual software development environments across VirtualBox, KVM, Hyper-V, Docker containers, VMware, and AWS, started as a personal side-project by Mitchell Hashimoto in January 2010 and first released in March 2010.

 In March 2010, the Apache Solr search server joined Lucene as a sub-project, merging the two developer communities.

 Flask is a micro web framework written in Python that does not require particular tools or libraries, created by Armin Ronacher of Pocoo, an international group of

Python enthusiasts, and first released on April 1, 2010.

🌐 WebSocket protocol development was transferred from the WHATWG/W3C to the IETF's HyBi Working Group in February 2010. A security vulnerability in the early draft — proxy cache poisoning — was discovered, causing several browsers to temporarily disable WebSocket support.

🌐 In April 2010, Steve Jobs published "Thoughts on Flash," an open letter explaining why Apple would not support Flash on iOS, citing security flaws, poor performance, proprietary control, and battery drain. The letter accelerated Flash's decline and is widely regarded as its "death blow."

🔑 OAuth is an open standard for access delegation commonly used by internet users to grant websites or applications access to their information on other websites without sharing passwords, with the OAuth 1.0 protocol published as RFC 5849 in April 2010.

💻 On 3 May 2010, it was announced that Stack Overflow had raised \$6 million in venture capital from a group of investors led by Union Square Ventures.

☁ Google Cloud Storage is an online file storage web service for storing and accessing data on Google Cloud Platform infrastructure, launched on May 19, 2010.

📊 BigQuery is a managed, serverless data warehouse product by Google offering scalable analysis over large quantities of data, originating from Google's internal Dremel technology that enabled quick queries across trillions of rows, launched on May 19, 2010.

⚙ Google released Android 2.2 Froyo on May 20, 2010, which brought significant performance improvements through the introduction of a Just-In-Time (JIT) compiler and added support for Wi-Fi hotspot functionality.

⚙ Apple rebranded the operating system from "iPhone OS" to "iOS" with the release of iOS 4 on June 21, 2010, which also introduced multitasking and home screen folders.

🎨 Sketch 1.0, a vector graphics editor for macOS conceived by Pieter Ommvlee to address frustrations with Adobe Illustrator's complexity for UI work, was released in private beta by Bohemian Coding on September 7, 2010, and went on to become the dominant interface-design tool through the mid-2010s.

📖 Jasmine 1.0 was released on September 14, 2010 by Pivotal Labs as a behaviour-driven development (BDD) testing framework for JavaScript, running in both browsers and Node.js without depending on a DOM. It was influenced by RSpec, JSpec, and JSSpec.

🔒 ZAP (Zed Attack Proxy) is a dynamic application security testing tool published

under the Apache License that allows users to manipulate all traffic passing through it, including HTTPS encrypted traffic. The first release was announced on Bugtraq in September 2010 and became an OWASP project a few months later.

🔗 Unicode 6.0 was released in October 2010, becoming the first version of the Unicode Standard to include standardized emoji characters, incorporating 722 emoji that had originated from Japanese mobile phone carriers.

📊 Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis, first released on October 1, 2010.

📦 NuGet is a package manager designed to enable developers to share reusable code as a software-as-a-service solution with a free, open-source client app, initially created by the Outercurve Foundation under the name NuPack. Originally distributed as a Visual Studio extension, starting with Visual Studio 2012, both Visual Studio and Visual Studio for Mac natively consume NuGet packages, with the project first released on October 5, 2010.

📖 RSpec 2.0 was released on October 10, 2010, refactoring the monolithic gem into separate components: `rspec-core` (runner and formatters), `rspec-expectations` (matchers), and `rspec-mocks` (mocking and stubbing), allowing projects to include only what they need.

📖 Pillow is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is a fork of the Python Imaging Library (PIL) and was first released on October 18, 2010.

🏢 Microsoft announced Office 365 on October 19, 2010, initially in a private beta, combining its existing cloud services like Office Web Apps, SharePoint Online, and Exchange Online into a single subscription-based service

🌐 AngularJS is a discontinued free and open-source JavaScript-based web framework for developing single-page applications, maintained mainly by Google and a community of individuals and corporations, first released on October 20, 2010.

🌐 Express.js is a backend web application framework for building RESTful APIs with Node.js, founded by TJ Holowaychuk and released as free and open-source software under the MIT License, with version 0.12 released on November 16, 2010.

☁️ Salesforce.com acquired Heroku on December 8, 2010, for approximately \$212 million in cash, making the cloud PaaS platform a wholly owned subsidiary.

🔒 JSON Web Token (JWT) is a proposed Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts a

number of claims, first published on December 28, 2010.

✂ In December 2010, the first documented description of QR code-based payments came from two patents filed by Shaun Cooley and Andrew Charles Payne, based on a prototype system developed for Norton Labs at Symantec called Norton Mobile Pay.

📖 "The RSpec Book: Behaviour-Driven Development with RSpec, Cucumber, and Friends" by David Chelimsky, Dave Astels, Zach Dennis, Aslak Hellesøy, Bryan Helmkamp, and Dan North was published by Pragmatic Bookshelf in December 2010, becoming the definitive guide to behaviour-driven development in Ruby and documenting the workflow that combines RSpec for unit-level specs with Cucumber for acceptance tests.

📖 Rust is a multi-paradigm, general-purpose programming language that enforces memory safety without requiring a garbage collector or reference counting, growing out of a personal project begun in 2006 by Mozilla employee Graydon Hoare. Mozilla began sponsoring the project in 2009 and officially announced it in 2010.

2011

💻 Stack Exchange publicly launched in January 2011 with 33 web sites; it had 27 employees and 1.5 million users at the time.

🏢 Apache Kafka is a distributed event store and stream-processing platform, an open-source system developed by the Apache Software Foundation written in Java and Scala, designed to provide a unified, high-throughput, low-latency platform for handling real-time data feeds, first released in January 2011.

☁ Jenkins is an open source automation server that helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. Originally named Hudson, the project was renamed to Jenkins in 2011 after a dispute with Oracle, first released on February 2, 2011.

⚙ Android 3.0 Honeycomb, a version specifically optimized for tablets, was released on February 22, 2011, introducing a new "Holographic" user interface and hardware acceleration.

⚙ The last unassigned top-level address blocks of 16 million IPv4 addresses were allocated in February 2011 by the IANA to the regional Internet registries, intensifying the push toward IPv6 adoption.

⚙ Chocolatey is a machine-level, command-line package manager and installer for Windows software. It was created by Rob Reynolds and first released on March 23,

2011 (v0.9.0). It was built on top of the NuGet infrastructure and inspired by the apt-get experience for Windows.

 Package Installer for Python (pip) is the de facto and recommended package management system written in Python for installing and managing software packages, first released on April 4, 2011.

 World IPv6 Day was a technical testing and publicity event on June 8, 2011, sponsored and organized by the Internet Society, in which major organizations including Google, Facebook, Yahoo, Akamai Technologies, and Limelight Networks enabled IPv6 on their main services for 24 hours.

 Office 365 reached general availability on June 28, 2011, initially targeting corporate users as a successor to the Microsoft Online Services Productivity Suite (BPOS).

 Time-based one-time password (TOTP) is a computer algorithm that generates a one-time password (OTP) that uses the current time as a source of uniqueness, officially becoming RFC 6238 in May 2011.

 In May 2011, Fedora Linux became the first major Linux distribution to enable systemd by default, replacing Upstart.

 Apache Flink is an open-source, unified stream-processing and batch-processing framework developed by the Apache Software Foundation. The core of Apache Flink is a distributed streaming data-flow engine written in Java and Scala, with initial release in May 2011.

 Selenium 2.0 was released in July 2011, merging the original Selenium RC with Simon Stewart's WebDriver project. WebDriver's native browser control APIs for Java, Python, Ruby, and C# became the new standard interface.

 Mac OS X 10.7 Lion was released on July 20, 2011. Apple shortened the product name from "Mac OS X" to "OS X" starting with this release, aligning the branding with its other operating systems.

 Linux 3.0 was released on July 21, 2011. The jump from 2.6 to 3.0 was primarily to celebrate the 20th anniversary of Linux and to simplify the version numbering, rather than signifying massive architectural changes.

 Kotlin is a cross-platform, statically typed, general-purpose programming language with type inference designed to interoperate fully with Java. The JVM version of Kotlin's standard library depends on the Java Class Library, but type inference allows its syntax to be more concise, first appearing on July 22, 2011.

 In July 2011, Heroku expanded its polyglot support by adding Node.js and Clojure

alongside the original Ruby, and Yukihiro Matsumoto, creator of Ruby, joined the company as Chief Architect for Ruby.

🌐 Google Native Client (NaCl) was a sandboxing technology for running either a subset of Intel x86, ARM, or MIPS native code, or a portable executable, in a sandbox, allowing safe execution of native code from a web browser independent of the user operating system and enabling web apps to run at near-native speeds in alignment with Google's plans for ChromeOS. It was initially released on September 16, 2011.

🏢 Apache Storm is a distributed stream processing computation framework written predominantly in the Clojure programming language. Originally created by Nathan Marz and team at BackType, the project was open sourced after being acquired by Twitter, with initial release on September 17, 2011.

🌐 Ruby on Rails 3.1 was released on August 31, 2011, introducing the Asset Pipeline, jQuery as the default JavaScript library, and support for CoffeeScript and Sass.

☁ GitLab was created in 2011 by Ukrainian programmer Dmytro Zaporozhets as a side project written in Ruby on Rails. The first code commit to the GitLab project was made on October 8, 2011.

⚙ Android 4.0 Ice Cream Sandwich was released on October 18, 2011, aiming to unify the phone and tablet branches of the operating system into a single platform with a refined "Holo" design language.

📖 Microsoft released a version of C# with `async/await` for the first time in the Async CTP (2011). And were later officially released in C# 5 (2012).

🏢 James Dixon, then chief technology officer at Pentaho, coined the term data lake by 2011 to contrast it with data mart, which is a smaller repository of interesting attributes derived from raw data.

☁ Google App Engine became generally available (GA) in November 2011.

📖 Mocha was created by TJ Holowaychuk and released in November 2011 as a flexible JavaScript test framework for Node.js and browsers, offering BDD and TDD interfaces and emphasising accurate reporting of asynchronous operations.

🌐 The WebSocket protocol was finalized and published by the IETF as RFC 6455 in December 2011, standardizing the full bidirectional communication protocol over a single TCP connection with a defined handshake, framing format, and security model.

🌐 Locust was created in 2011 by Jonatan Heyman and Carl Bystrom at ESN while load testing Battlelog, a companion app for Battlefield 3. Unable to find a suitable tool for simulating realistic user behaviour, they built Locust in Python using `gevent`, allowing a

single process to handle thousands of concurrent users.

🎨 InVision, a cloud-based prototyping and collaboration platform that turned static Sketch and Photoshop screens into clickable prototypes, was founded by Clark Valberg and Ben Nadel in 2011 and became an industry standard for design handoff, reaching a peak valuation of \$1.9 billion.

2012

📖 The first numbered pre-alpha version of the compiler, Rust 0.1, was released in January 2012.

🌐 Gatling was first released on January 13, 2012 by Stéphane Landelle as an open-source load and performance testing framework built on Scala and Netty, treating performance tests as production code with a DSL-based scenario definition.

📖 "The Cucumber Book: Behaviour-Driven Development for Testers and Developers" by Matt Wynne and Aslak Hellesøy was published by Pragmatic Bookshelf in January 2012, serving as the definitive guide to Cucumber and showing how to write executable acceptance tests in the plain-language Gherkin syntax to bridge communication between business stakeholders and developers.

☁️ Ansible was written by Michael DeHaan and acquired by Red Hat in 2015, with initial release on February 20, 2012.

🐛 A "leap day bug" caused a significant service disruption for Windows Azure users on February 29, 2012.

⚙️ The Raspberry Pi is a small single-board computer that runs Linux-based operating systems. It was designed by the Raspberry Pi Foundation and first released on February 29, 2012 (announcement date; availability began in May 2012) as an affordable platform for promoting computer science education. The original Model B featured an ARM processor, 256 MB RAM, and cost \$35, making it an accessible tool for hobbyists, educators, and developers worldwide.

📖 Go 1.0 was released on March 28, 2012, as the first stable version, accompanied by a long-term backward-compatibility guarantee for the Go 1 language specification.

📖 On April 24, 2012, José Valim published "A peek inside Elixir's Parallel Compiler," signaling the early development and design goals of the language at Plataformatec.

☁️ Google Drive was introduced on April 24, 2012, as a cloud storage and synchronization service that unified file management across the Google Apps suite.

☁️ Microsoft announced major support for Virtual Machines (including Linux), Web

Sites, and a Python SDK for Windows Azure in June 2012.

☁ Google Compute Engine (GCE) is the infrastructure as a service (IaaS) component of Google Cloud Platform built on the global infrastructure that runs Google's search engine, Gmail, YouTube and other services. Google announced Compute Engine on June 28, 2012 at Google I/O 2012 in a limited preview mode.

📖 TypeScript is a free and open source programming language developed and maintained by Microsoft, first made public on October 1, 2012 (at version 0.8), after two years of internal development at Microsoft.

🌐 QUIC is a general-purpose transport layer network protocol initially designed by Jim Roskind at Google, implemented, and deployed in 2012, with formal introduction on October 12, 2012.

📊 Apache Lucene version 4.0 was released on October 12, 2012, introducing major features such as pluggable codecs and near-real-time search improvements.

📊 Amazon Redshift is a data warehouse product forming part of the larger cloud-computing platform Amazon Web Services, built on top of technology from the massive parallel processing (MPP) data warehouse company ParAccel (later acquired by Actian) to handle large scale data sets and database migrations. It was initially released in October 2012.

🌐 Internet Explorer 10, released in October 2012 with Windows 8, introduced WebSocket support for Windows users, completing mainstream browser adoption alongside Firefox 4 (2011) and Safari 5 (2010).

🔒 OAuth 2.0 was published as RFC 6749 in October 2012.

🌐 Emscripten is an LLVM/Clang-based compiler that compiles C and C++ source code to WebAssembly (or to a subset of JavaScript known as asm.js, its original compilation target before the advent of WebAssembly in 2017), primarily for execution in web browsers, with first release on November 11, 2012.

🔒 HTTP Strict Transport Security (HSTS) is a policy mechanism that helps protect websites against man-in-the-middle attacks such as protocol downgrade attacks and cookie hijacking. The HSTS specification was published as RFC 6797 on November 19, 2012, after being approved on October 2, 2012 by the IESG for publication as a Proposed Standard RFC.

☁ GitLab CI, a continuous integration tool created by Dmitriy Zaporozhets, was first released as a standalone service on November 22, 2012. It was later integrated into the core GitLab application in September 2015, forming a unified DevOps platform.

🌐 In December 2012, W3C designated HTML5 as a Candidate Recommendation, signaling the specification was feature-complete and entering final review. WHATWG simultaneously began maintaining HTML as a continuously updated "Living Standard" without version numbers.

🏢 From 2012, Microsoft became a significant user of GitHub, using it to host open-source projects and development tools such as .NET Core, Chakra Core, MSBuild, PowerShell, PowerToys, Visual Studio Code, Windows Calculator, Windows Terminal and the bulk of its product documentation (now to be found on Microsoft Docs).

☁ Prometheus is a free software application used for event monitoring and alerting that records real-time metrics in a time series database built using a HTTP pull model. Prometheus was developed at SoundCloud starting in 2012 when the company discovered that its existing metrics and monitoring solutions (using StatsD and Graphite) were not sufficient for their needs.

🏢 Elastic NV was founded in 2012 to provide commercial services and products around Elasticsearch and related software.

👤 Dylan Field and Evan Wallace began working on Figma in 2012 while studying computer science at Brown University; Field was named a Thiel Fellow the same year, earning him \$234,460 in exchange for taking a leave of absence from college.

2013

📖 A Ruby implementation of AsciiDoc called 'Asciidoctor' was released on January 30, 2013, and is in use by GitHub and GitLab.

📖 TOML is a file format for configuration files intended to be easy to read and write due to obvious semantics which aim to be "minimal," and is designed to map unambiguously to a dictionary, with initial release on February 23, 2013.

📖 Ruby 2.0 was intended to be fully backward compatible with Ruby 1.9.3, with the official 2.0.0 release on February 24, 2013.

💻 Meson is a software tool for automating the building (compiling) of software, with the overall goal of promoting programmer productivity. Meson is free and open-source software written in Python under the Apache License 2.0, with initial release on March 2, 2013.

👤 *Peopleware: Productive Projects and Teams* was revised for its 3rd Edition in 2013

🏢 Apache Parquet was created by Twitter and Cloudera and first announced in March 2013.

☁ Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Docker was introduced to the public by Solomon Hykes at PyCon in Santa Clara on March 20, 2013, and released as an open-source project. At the time, it used LXC as its default execution environment.

🏢 On October 29, 2013, dotCloud, Inc. officially rebranded as Docker, Inc. to reflect the project's massive growth and shift in company focus.

🌐 asm.js is a subset of JavaScript designed to allow computer software written in languages such as C to be run as web applications while maintaining performance characteristics considerably better than standard JavaScript, first appearing on March 21, 2013.

🌐 The W3C published the WebSocket API specification in 2013, defining the JavaScript interface (`new WebSocket()`) separately from the underlying RFC 6455 protocol, completing the full standardization of WebSocket as a web platform feature.

🏢 On April 15, 2013, it was announced that the Xen Project was moved under the auspices of the Linux Foundation as a Collaborative Project.

🔒 Starting with Java SE 7 Update 21 in April 2013, unsigned Java Applets began displaying security warnings; from Java 7 Update 51, unsigned applets were blocked by default, marking the start of a security-driven decline of the technology.

⚙️ Google Compute Engine was released to general availability in May 2013.

📞 An expired SSL certificate caused a widespread outage for Windows Azure and Xbox Live on February 22, 2013.

📖 Java EE 7, released on May 28, 2013, added support for WebSocket, JSON processing, and HTML5, modernizing the platform for the web.

🌐 React is a free and open-source front-end JavaScript library for building user interfaces based on UI components, maintained by Meta (formerly Facebook) and a community of individual developers and companies, with initial release on May 29, 2013.

📊 Plotly is an interactive, open-source, and browser-based graphing library for Python. The company was founded in 2012, and the Python library was first released in June 2013.

📊 Seaborn is a Python data visualization library based on matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. It was first released in December 2013.

⚙️ The IEEE 802.11ac amendment (later branded Wi-Fi 5) was published in December 2013.

🌐 Electron (formerly known as Atom Shell) is a free and open-source software framework developed and maintained by GitHub, designed to create desktop applications using web technologies rendered using a flavor of the Chromium browser engine and a backend using the Node.js runtime environment. It was initially released on July 15, 2013.

🌐 The QUIC code was experimentally developed in Google Chrome starting in 2012, and was announced as part of Chromium version 29 (released on August 20, 2013).

📊 Power BI, a suite of business intelligence and self-service data visualization tools, was unveiled for Office 365 on July 8, 2013, allowing users to analyze and visualize data within Excel

🗄️ Slack, which stands for "Searchable Log of All Conversation and Knowledge," was officially launched to the public in August 2013 and gained 8,000 signups within its first 24 hours.

🌐 In September 2013, Google announced it would phase out NPAPI support in Chrome during 2014, calling its "90s-era architecture" a leading cause of hangs, crashes, and security incidents. This triggered a cascade of NPAPI removal across all major browsers, effectively ending Java Applet support in modern browsers.

📊 InfluxDB is an open-source time series database (TSDB) developed by the company InfluxData, with initial release on September 24, 2013.

🌐 In October 2013, Ecma International published the first edition of its JSON standard ECMA-404.

📊 Presto is a distributed query engine for big data using the SQL query language whose architecture allows users to query data sources such as Hadoop, Cassandra, Kafka, AWS S3, Alluxio, MySQL, MongoDB and Teradata, and allows use of multiple data sources within a query. Hive was deemed too slow for Facebook's scale and Presto was invented to fill the gap to run fast queries, with original development starting in 2012 and deployment at Facebook later that year. It was initially released on November 10, 2013.

🗄️ On November 13, 2013, Microsoft announced the release of a software as a service offering of Visual Studio on Microsoft Azure platform; at the time, Microsoft called it Visual Studio Online.

📊 Amazon Kinesis is a family of services provided by Amazon Web Services (AWS) for processing and analyzing real-time streaming data at a large scale. Launched in

November 2013.

⚙️ In 2013, adequate containers support functionality was finished in kernel version 3.8 with the introduction of User namespaces.

🌐 MEAN (MongoDB, Express.js, AngularJS (or Angular), and Node.js) is a free and open-source JavaScript software stack for building dynamic web sites and web applications. A variation known as MERN replaces Angular with React. The acronym MEAN was coined by Valeri Karpov. He introduced the term in a 2013 blog post.

🏢 The FIDO ("Fast IDentity Online") Alliance is an open industry association launched in February 2013 whose stated mission is to develop and promote authentication standards that "help reduce the world's over-reliance on passwords".

🔒 The Adversarial Tactics, Techniques, and Common Knowledge or MITRE ATT&CK is a guideline for classifying and describing cyberattacks and intrusions. It was created by the Mitre Corporation and released in 2013.

🏢 DevOps Research and Assessment (abbreviated to DORA) is a team that is part of Google Cloud that engages in opinion polling of software engineers to conduct research for the DevOps movement. The DORA team began publishing State of DevOps Reports in 2013.

2014

⚙️ FreeBSD 10.0-RELEASE was released on January 20, 2014. In this version, Clang/LLVM replaced GCC as the default system compiler, and the bhyve hypervisor was introduced.

💻 MkDocs is a static site generator designed for building project documentation, written in Python and also used in other environments. It converts Markdown files into HTML pages, effectively creating a static website containing documentation, with initial release on January 24, 2014.

🌐 In January 2014 CORS was accepted as a W3C Recommendation.

🌐 Webpack is a free and open-source module bundler for JavaScript made primarily for JavaScript, but capable of transforming front-end assets such as HTML, CSS, and images if the corresponding loaders are included, with initial release on February 19, 2014.

🔒 OpenID Connect is the third generation of OpenID technology, published in February 2014 by the OpenID Foundation. It is an authentication layer on top of the OAuth 2.0 authorization framework.

⚙️ In February 2014, Mark Shuttleworth announced that Ubuntu would use systemd to replace its old Upstart init system following Debian's decision. This led to a "widely publicized debate" within the Debian Technical Committee, eventually resulting in the resignation of several prominent developers due to the stress of the dispute.

⚙️ eBPF (Extended BPF) was merged into the Linux kernel in March 2014 with version 3.15. It is the successor to the Berkeley Packet Filter (BPF) and extended it with a more sophisticated instruction set and the ability to attach programs to various kernel hooks beyond networking, such as tracepoints and kprobes.

🏢 The .NET Foundation is an organization incorporated on March 31, 2014, by Microsoft to improve open-source software development and collaboration around the .NET Framework.

📖 TypeScript 1.0 was released on April 12, 2014 at Microsoft's Build developer conference, with Visual Studio 2013 Update 2 providing built-in support for TypeScript.

☁️ In April 2014, Windows Azure renamed Microsoft Azure.

☁️ Microsoft introduced the Azure Resource Manager (ARM) portal at the Build 2014 conference in April 2014.

🐛 Heartbleed is a security bug in some outdated versions of the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security (TLS) protocol. It was introduced into the software in 2012 and publicly disclosed in April 2014.

⚙️ In late April 2014, a campaign to boycott systemd was launched by critics who argued its architecture "violates the Unix philosophy" and suffers from "feature creep."

📊 Apache Spark is an open-source unified analytics engine for large-scale data processing that provides an interface for programming clusters with implicit data parallelism and fault tolerance. Spark was initially started by Matei Zaharia at UC Berkeley's AMPLab in 2009 and open sourced in 2010 under a BSD license. In 2013, the project was donated to the Apache Software Foundation and switched its license to Apache 2.0, with the first version (v1.0) released on May 26, 2014.

💻 Git 2.0 was released on May 28, 2014.

📖 RSpec 3.0 was released on June 1, 2014, introducing verified doubles in rspec-mocks (enforcing method existence on test doubles) and composable matchers in rspec-expectations, while removing features deprecated in 2.x.

🌐 Cypress was founded in 2014 by Brian Mann as a reaction to the limitations of Selenium. The first code commit was made on June 5, 2014, and the project entered

private beta in 2015 after approximately 18 months of development to reach MVP.

📖 Jest was open-sourced by Facebook in June 2014, having been developed internally since around 2011 for testing the web chat application. It introduced automatic module mocking and parallel test execution, and became the dominant testing framework for React applications.

☁ Docker 1.0 was released on June 9, 2014, marking the platform as "production-ready" for enterprise and production environments.

☁ Kubernetes was officially announced as an open-source project by Google at DockerCon on June 10, 2014. The project was created by Joe Beda, Brendan Burns, and Craig McLuckie, drawing inspiration from Google's internal cluster management systems, Borg and Omega.

👤 Google unveiled Material Design, a cross-platform visual design language, at Google I/O on June 25, 2014.

☁ Terraform is an open-source, infrastructure as code software tool created by HashiCorp, with initial release on July 28, 2014.

📖 Elixir v1.0 was released on September 18, 2014, establishing the language's stability and readiness for production use on the Erlang VM (BEAM).

🐛 Shellshock, also known as Bashdoor, is a family of security bugs in the Unix Bash shell, the first of which was disclosed on 24 September 2014. Shellshock could enable an attacker to cause Bash to execute arbitrary commands and gain unauthorized access to many Internet-facing services, such as web servers, that use Bash to process requests.

🌐 Babel is a free and open-source JavaScript transpiler mainly used to convert ECMAScript 2015+ (ES6+) code into a backwards compatible version of JavaScript that can be run by older JavaScript engines, with initial release on September 28, 2014.

🌐 On October 28, 2014, HTML5 was published as a W3C Recommendation.

🧠 On November 6, 2014, Amazon announced Alexa alongside the Echo, a smart speaker device developed by Amazon Lab126.

📖 On November 12, 2014, Microsoft announced .NET Core, in an effort to include cross-platform support for .NET, including Linux and macOS, and the adoption of a conventional ("bazaar"-like) open-source development model under the stewardship of the .NET Foundation.

⚙ Android 5.0 Lollipop was released on November 12, 2014, introducing a major

redesign of the user interface based on the "Material Design" language, as well as the Android Runtime (ART) replacing the Dalvik virtual machine for improved performance.

☁ On November 13, 2014, AWS launches a preview of EC2 Container Service (ECS), facilitating the use of container infrastructure on AWS. Third-party integration such as those with Docker are available at the time of release.

☁ On November 13, 2014, AWS launches AWS Lambda, its Functions as a Service (FaaS) tool. With Lambda, AWS customers can define and upload functions with specific triggers and execution code.

🔧 A storage upgrade caused a major global outage for Microsoft Azure on November 18, 2014.

🔒 Let's Encrypt is a non-profit certificate authority run by Internet Security Research Group (ISRG) that provides X.509 certificates for Transport Layer Security (TLS) encryption at no charge. Let's Encrypt was announced publicly on November 18, 2014.

📊 PostgreSQL 9.4 was released on December 18, 2014, introducing the JSONB data type for efficient storage and indexing of semi-structured data.

🌐 HTML5 was published as a W3C Recommendation on October 28, 2014, completing nearly a decade of development and establishing the standard for the modern web with native support for video, audio, canvas, local storage, and semantic elements.

☁ The alpha release of Google Kubernetes Engine (GKE, initially called Google Container Engine) was announced in November 2014.

🏢 In 2014, Satya Nadella was named the new CEO of Microsoft. Microsoft began to adopt open source into its core business. In contrast to Ballmer's stance, Nadella presented a slide that read, "Microsoft loves Linux".

☁ In 2014, with the release of version 0.9, Docker replaced LXC with its own component, libcontainer, which was written in the Go programming language.

⚙️ OverlayFS is a union mount filesystem implementation for Linux. It combines multiple different underlying mount points into one, resulting in single directory structure that contains underlying files and sub-directories from all sources. It was merged into the Linux kernel mainline in 2014, in kernel version 3.18. It was improved in version 4.0, bringing improvements necessary for e.g. the overlay2 storage driver in Docker.

☁ Grafana is a multi-platform open source analytics and interactive visualization web application that provides charts, graphs, and alerts for the web when connected to supported data sources. Grafana was first released in 2014 by Torkel Ödegaard as an

offshoot of a project at Orbitz, targeting time series databases such as InfluxDB, OpenTSDB, and Prometheus. The Grafana user interface was originally based on version 3 of Kibana.

🧠 seq2seq is an approach to machine translation (or more generally, sequence transduction) with roots in information theory, where communication is understood as an encode-transmit-decode process and machine translation can be studied as a special case of communication. The idea of encoder-decoder sequence transduction had been developed in the early 2010s, with the papers most commonly cited as the originators that produced seq2seq from two papers in 2014. In the seq2seq as proposed by them, both the encoder and the decoder were LSTMs, which had the "bottleneck" problem. The attention mechanism, proposed in 2014, resolved this bottleneck problem.

💻 The Neovim project was started in 2014 by Thiago de Arruda as an effort to completely rework the outdated Vim source code, with a successful fundraising campaign in March 2014 that supported at least one full-time developer.

Timeline - 2015-19

2015

🏢 By 2013, Prometheus was introduced for production monitoring at SoundCloud. The official public announcement was made in January 2015.

📊 Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages, spun off from IPython in 2014 by Fernando Pérez and Brian Granger, and formally established in February 2015.

🧠 Keras is an open-source software library that provides a Python interface for artificial neural networks and acts as an interface for the TensorFlow library, with initial release on March 27, 2015.

💻 On April 8, 2015, GitHub announced Git Large File Storage (Git LFS), which allows users to store and work with large binary files in Git.

💻 Visual Studio Code, a source-code editor developed by Microsoft for Windows, Linux, and macOS, was first announced on April 29, 2015, at the Build conference, with a preview build released shortly after.

⚙️ In April 2015, Debian 8.0 (Jessie) was released with systemd as the default init system. On the same month, Ubuntu 15.04 was also released, officially switching to systemd.

🌐 Discord was publicly released on May 13, 2015, as a platform designed specifically for the gaming community to communicate while playing together.

📖 The first stable release, Rust 1.0, was announced on May 15, 2015.

📖 The 6th edition, ECMAScript 6 (ES6) and later renamed to ECMAScript 2015, was finalized in June 2015, adding significant new syntax for writing complex applications including class declarations and ES6 modules. ES6 supports "arrow function" syntax, where a $\Rightarrow$ symbol separates the anonymous function's parameter list from the body.

☁ The Open Container Initiative (OCI) was launched at DockerCon on June 22, 2015, by Docker, CoreOS, and other industry leaders under the Linux Foundation to design open standards for operating-system-level virtualization. OCI develops runc, a container runtime that implements their specification and serves as a basis for other higher-level tools. runc was first released in July 2015 as version 0.0.1.

🧠 Amazon Echo and Alexa had their full U.S. launch on June 28, 2015, following the limited November 2014 release.

🏢 Kubernetes 1.0 was released on July 21, 2015. Google worked with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF) and offer Kubernetes as a seed technology.

⚙ Windows 10, released on July 29, 2015, introduced the "Windows as a Service" model with continuous updates, restored the Start menu after the tile-based experiment of Windows 8, and integrated the Microsoft Edge browser and the Cortana virtual assistant.

📖 Go 1.5 was released on August 19, 2015. This was a landmark release in which the entire compiler and runtime toolchain was rewritten from C into Go, making the language self-hosting.

🌐 Phoenix v1.0 was released on September 1, 2015, providing a productive and reliable web framework that became a cornerstone of the Elixir ecosystem.

🔒 The FIDO2 Project is a joint effort between the FIDO Alliance and the World Wide Web Consortium (W3C) whose goal is to create strong authentication for the web. At its core, FIDO2 consists of the W3C Web Authentication (WebAuthn) standard and the FIDO Client to Authenticator Protocol 2 (CTAP2), with the FIDO 2.0 Proposed Standard released on September 4, 2015.

🏢 On September 14, 2015, Let's Encrypt issued its first certificate, which was for the domain helloworld.letsencrypt.org. On the same day, ISRG submitted its root program applications to Mozilla, Microsoft, Google and Apple.

🏠 In September 2015, Node.js v0.12 and io.js v3.3 were merged back together into Node v4.0.

🌐 Vue.js is an open-source model-view-viewmodel front end JavaScript framework for building user interfaces and single-page applications, created by Evan You. The first source code commit to the project was dated July 2013, and Vue was first released in February 2014, with Vue.js 1.0 released on October 27, 2015.

☁ The Serverless Framework is a free and open-source web framework written using Node.js and the first framework developed for building applications on AWS Lambda. Developed by Austen Collins and maintained by a full-time team, it was initially released in October 2015.

💻 Neovim's first official release occurred on November 1, 2015, delivering a refactored, extensible fork of Vim focused on modern features and embeddability.

🧠 TensorFlow is a free and open-source software library for machine learning and artificial intelligence usable across a range of tasks but with a particular focus on training and inference of deep neural networks, with initial release on November 9, 2015.

🏠 On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub.

🏠 On November 18, 2015, Microsoft announced that Visual Studio Online was rebranded as "Visual Studio Team Services (VSTS)".

📖 Python added support for async/await with version 3.5 in 2015.

📖 TypeScript added support for async/await with version 1.7 in 2015.

🏠 The Cloud Native Computing Foundation (CNCF) is a Linux Foundation project that was founded in 2015 to help advance container technology and align the tech industry around its evolution.

🌐 HTTP/2, published in 2015, provides a more efficient expression of HTTP's semantics "on the wire".

☁ Envoy, a high-performance open source edge and service proxy that makes the network transparent to applications, was originally built at Lyft to move their architecture away from a monolith. Development began in 2015, and by late fall of 2015 Envoy was serving 100% of Lyft traffic.

2016

☁ In February 2016 Helm package manager for Kubernetes was released.

☁ Google Cloud Functions, a serverless execution environment, became available in Alpha in February 2016.

☁ Azure IoT Hub reached general availability on February 4, 2016, providing a central message hub for bi-directional communication between IoT applications and devices.

☁ Kubernetes 1.2, released in March 2016, introduced Deployments and support for clusters with up to 1,000 nodes.

⚙ Azer Koçulu's **left-pad**, an 11-line JavaScript utility package published to npm on August 15, 2014, had come to be downloaded over 15 million times a month and depended upon by major projects including Babel, Webpack, and React; after a dispute in which npm transferred ownership of his **kik** package name to Kik Interactive on March 18, 2016, Koçulu unpublished all 273 of his npm packages in protest on March 22, 2016, breaking the build of thousands of projects across the JavaScript ecosystem, including those of Facebook, PayPal, Netflix, and Spotify, within hours.

⚙ npm restored the **left-pad** package by manually republishing version 0.0.3 from a backup a few hours after it was unpublished on March 22, 2016; the company subsequently changed its policy to prevent unpublishing a package once other packages depend on it.

💻 Visual Studio Code graduated from its public preview stage and reached version 1.0 on April 14, 2016, at which point it had over 1,000 extensions and half a million active users.

🔒 The General Data Protection Regulation, abbreviated GDPR, or French RGPD is a European Union regulation on information privacy in the European Union (EU) and the European Economic Area (EEA). The European Parliament and Council of the European Union adopted the GDPR on 14 April 2016, to become effective on 25 May 2018.

☁ On April 20, 2016, Jenkins version 2 was released with the Pipeline plugin enabled by default. The plugin allows for writing build instructions using a domain specific language based on Apache Groovy.

⚙ Snap is a software packaging and deployment system developed by Canonical for operating systems that use the Linux kernel and the systemd init system. The first snapd 2.12 is released into Ubuntu 16.04 (xenial-updates) and Fedora 24.

🔒 Web Authentication (WebAuthn) is a web standard published by the World Wide Web Consortium (W3C). WebAuthn is a core component of the FIDO2 Project under the

guidance of the FIDO Alliance. It was first published on 31 May 2016.

 In May 2016, the Cloud Native Computing Foundation accepted Prometheus as its second incubated project, after Kubernetes.

 ASP.NET Core is a free and open-source web framework and successor to ASP.NET, developed by Microsoft, with initial release on June 7, 2016.

 Yarn is a software packaging system developed in 2016 by Facebook for the Node.js JavaScript runtime environment, with initial release on June 18, 2016.

 .NET Core 1.0 was released on June 27, 2016, along with Microsoft Visual Studio 2015 Update 3, which enables .NET Core development.

 The Language Server Protocol (LSP) is an open, JSON-RPC-based protocol for use between source code editors or integrated development environments (IDEs) and servers that provide programming language-specific features, originally developed for Microsoft Visual Studio Code and now an open standard. On June 27, 2016, Microsoft announced a collaboration with Red Hat and Codenvy to standardize the protocol's specification.

 The VS Code Go extension provides rich language support for the Go programming language, with initial release in July 2016.

 Prometheus 1.0 was released in July 2016. Subsequent versions were released through 2016 and 2017, leading to Prometheus 2.0 in November 2017.

 Windows Subsystem for Linux (WSL) is a compatibility layer for running Linux binary executables (in ELF format) natively on Windows. WSL beta was introduced in Windows 10 version 1607 (Anniversary Update) on August 2, 2016. Only Ubuntu (with Bash as the default shell) was supported.

 Microsoft announced that PowerShell was becoming open-source (under the MIT License) and cross-platform on August 18, 2016, with alpha versions released for Linux and macOS.

 Lyft open sourced Envoy on September 14, 2016, announcing it as a C++ L7 proxy and communication bus designed for large service oriented architectures, having tagged version 1.0.0 two days earlier.

 macOS 10.12 Sierra was released on September 20, 2016. Apple renamed the operating system from "OS X" to "macOS" to align with the branding of its other operating systems (iOS, watchOS, tvOS). It also introduced Siri on the Mac for the first time.

☁ Cilium 0.1, a cloud-native project that leverages eBPF to provide high-performance networking, security, and observability for containerized workloads, was released in September 2016.

🌐 Angular is a TypeScript-based free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular 2.0 was announced at the ng-Europe conference on October 22–23, 2014, with drastic changes in the 2.0 version creating considerable controversy among developers. Version 2.0 was released on September 14, 2016.

🧠 PyTorch is a machine learning framework based on the Torch library used for applications such as computer vision and natural language processing, originally developed by Meta AI and now part of the Linux Foundation umbrella, with initial release in September 2016.

🎨 Figma saw its first public release on September 27, 2016, launching as a browser-based collaborative interface design tool.

📁 Google Apps was officially rebranded as G Suite on September 29, 2016, to emphasize the collaborative and intelligent nature of its cloud-based productivity tools

🌐 Next.js is an open-source web development framework created by Vercel enabling React-based web applications with server-side rendering and generating static websites, with initial release on October 25, 2016.

🌐 Nuxt.js is a free and open source JavaScript library based on Vue.js, Node.js, Webpack and Babel.js, inspired by Next.js, which is a framework of similar purpose based on React.js, with initial release on October 26, 2016.

🌐 HTML 5.1 was published as a W3C Recommendation in November 2016, adding features such as the `<picture>` element for responsive images, the `<details>` and `<summary>` elements, and the `inputmode` attribute.

📁 On November 15, 2016, Microsoft announced the general availability of Azure Functions.

📁 Hugging Face, Inc. is an American company based in New York City that develops computation tools for building applications using machine learning. The company was founded in 2016, originally as a company that developed a chatbot app targeted at teenagers. The Hugging Face Hub is a platform serving as a centralized web service for hosting.

2017

🌐 k6 was open-sourced by Load Impact in February 2017 as a modern load testing tool written in Go with JavaScript scripting, designed for developer workflows and CI/CD integration. Load Impact was subsequently rebranded as k6 in 2020 and acquired by Grafana Labs in 2021.

🌐 WebAssembly (sometimes abbreviated Wasm) defines a portable binary-code format and a corresponding text format for executable programs. It was announced in 2015 and first released in March 2017.

🏢 Google acquired Kaggle, the world's largest community of data scientists and machine learning practitioners, in March 2017.

☁️ Kubernetes 1.6 was released in March 2017, introducing Role-Based Access Control (RBAC) and making etcd v3 the default storage backend.

🧠 DVC is a free and open-source, platform-agnostic version system for data, machine learning models, and experiments, designed to make ML models shareable, experiments reproducible, and to track versions of models, data, and pipelines, with initial release on May 4, 2017.

🏢 Azure Cosmos DB, a globally distributed multi-model database service, reached general availability on May 10, 2017.

☁️ On May 24, 2017, Google, IBM, and Lyft announced the first public release of Istio, an open source project that provides a uniform way to connect, secure, manage and monitor microservices. It was built using the Envoy proxy from Lyft, and gives visibility and control over traffic without requiring any changes to application code.

☁️ Google Cloud IoT Core was launched in Beta in May 2017 for connecting and managing IoT devices.

🐛 The WannaCry ransomware attack was a worldwide cyberattack in May 2017 by the WannaCry ransomware cryptoworm, which targeted computers running the Microsoft Windows operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency.

☁️ Kubernetes 1.7, released in June 2017, introduced Custom Resource Definitions (CRDs) as a replacement for Third Party Resources (TPRs).

🏢 The "Microsoft 365" brand was first introduced on July 10, 2017, at Microsoft Inspire as an enterprise subscription product that bundled Office 365 with Windows 10 Enterprise licenses and Enterprise Mobility + Security (EMS)

⚙️ On July 14, 2017, IPv6 was ratified as an Internet Standard with the publication of RFC 8200, which superseded RFC 2460.

☁️ OCI v1.0 was officially released on July 19, 2017, marking the stabilization of the Runtime and Image Specifications. runc 1.0, the reference implementation of the OCI Runtime Specification, was released on the same day.

🌐 In July 2017, Adobe announced its intention to discontinue Flash Player entirely — including security updates — by the end of 2020, prompting all major browsers and platforms to finalize their removal plans.

📊 Apache Iceberg is an open-source high-performance format for huge analytic tables that enables the use of SQL tables for big data while making it possible for engines like Spark, Trino, Flink, Presto, Hive, Impala, StarRocks, Doris, and Pig to safely work with the same tables at the same time, with initial release on August 10, 2017.

📖 On September 12, 2017, Oracle Corporation announced that it would submit Java EE to the Eclipse Foundation, where it was later renamed Jakarta EE because Oracle retained the "Java" trademark.

☁️ Lyft contributed Envoy to the Cloud Native Computing Foundation on September 13, 2017, where it was accepted at the incubating maturity level.

🌐 Java 9 was released on September 21, 2017, officially deprecating the Java Applet API (`java.applet` package), signaling the formal end of applet support within the Java platform.

🌐 Cypress announced it was going open source on May 4, 2017, and entered public beta on October 10, 2017. It was restructured into a single monorepo and made available as a Node.js module installable via npm, ending three years of private development.

⚙️ WSL was no longer beta in Windows 10 version 1709 (Fall Creators Update), released on October 17, 2017. Multiple Linux distributions could be installed and were available for install in the Windows Store.

⚙️ Ubuntu 17.10 (Artful Aardvark) was released on 19 October 2017, marking the first release to use the GNOME Shell interface and replace the X11 windowing system with the Wayland display server by default. This change occurred after nearly 6.5 years of using Unity as the default desktop.

📊 At the end of October 2017 Microsoft announced a preview of AKS (Azure Container Service), a managed Kubernetes service in Azure.

⚙️ On November 6, 2017, Amazon announced the new C5 family of instances that were

based on a custom architecture around the KVM hypervisor, called Nitro.

🧠 Amazon SageMaker is a cloud machine-learning platform that enables developers to create, train, and deploy machine-learning (ML) models in the cloud and also enables developers to deploy ML models on embedded systems and edge-devices. It was launched on November 29, 2017.

☁ Google Kubernetes Engine (GKE) became certified by the CNCF in November 2017, ensuring compatibility with the broader ecosystem.

🏢 Databricks grew out of the AMPLab project at University of California, Berkeley that was involved in making Apache Spark, an open-source distributed computing framework built atop Scala. In November 2017, the company was announced as a first-party service on Microsoft Azure via integration Azure Databricks.

☁ Kata Containers, an open-source project providing lightweight virtual machines that feel and perform like containers, was announced on December 5, 2017. It was formed as a merger of Intel Clear Containers and Hyper.sh runV.

🌐 HTML 5.2 was published as a W3C Recommendation in December 2017, adding support for the `<dialog>` element and the Payment Request API, while removing obsolete features.

📄 ISO published PDF 2.0 as ISO 32000-2 in 2017, the first version of the PDF specification developed entirely by ISO without Adobe proprietary extensions, introducing new features such as improved digital signatures, associated files, and better support for geospatial data.

🌐 WebAssembly System Interface (WASI) is a simple interface (ABI and API) designed by Mozilla intended to be portable to any platform. It provides POSIX-like features like file I/O constrained by capability-based security.

📄 Javascript added support for async/await in 2017 as part of ECMAScript 2017 JavaScript edition.

🧠 A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV). Transformers were introduced in 2017 by a team at Google Brain and are increasingly becoming the model of choice for NLP problems, replacing RNN models such as long short-term memory (LSTM).

🌐 Uvicorn, a lightning-fast ASGI (Asynchronous Server Gateway Interface) server implementation for Python, was created by Tom Christie and first released in late 2017.

🌐 Puppeteer was open-sourced by Google's Chrome DevTools team in 2017 as a Node.js library providing a high-level API over the Chrome DevTools Protocol (CDP) to control headless Chrome and Chromium. Puppeteer 1.0 was released in January 2018 and became the dominant tool for headless browser automation in the JavaScript ecosystem, later inspiring the creation of Playwright.

2018

💻 PowerShell Core 6.0 reached General Availability on January 10, 2018; built on .NET Core, it was the first version of the shell designed for cross-platform use on Windows, macOS, and Linux.

☁️ On January 15, 2018, AWS announced Go as a supported language for AWS Lambda

🔧 Meltdown is one of the two original transient execution CPU vulnerabilities (the other being Spectre). Meltdown affects Intel x86 microprocessors, IBM POWER processors, and some ARM-based microprocessors. It allows a rogue process to read all memory, even when it is not authorized to do so. It was discovered in January 2018.

🔧 Spectre refers to one of the two original transient execution CPU vulnerabilities (the other being Meltdown), which involve microarchitectural timing side-channel attacks. These affect modern microprocessors that perform branch prediction and other forms of speculation. It was discovered in January 2018.

⚙️ OpenSSH-based client and server programs have been included in Windows 10 since version 1803 (April 2018 Update)

💻 Python extension for Visual Studio Code is a Visual Studio Code extension with rich support for the Python language, with initial release on February 2, 2018.

☁️ Argo CD, a declarative, GitOps continuous delivery tool for Kubernetes, was started at Intuit as a Kubernetes-native solution fitting the needs of a large enterprise. Its repository was created on February 9, 2018, and version 0.1.0 was released on March 13, 2018.

🌐 Nest (NestJS) is a framework for building efficient, scalable Node.js server-side applications, with initial release on GitHub on February 16, 2018.

🧠 Cloud TPUs (Tensor Processing Units) became available in Beta in February 2018, offering hardware acceleration for machine learning.

📖 Deno is a runtime for JavaScript, TypeScript, and WebAssembly based on the V8 JavaScript engine and the Rust programming language, co-created by Ryan Dahl, who also created Node.js, with initial release on May 13, 2018.

☁ Google One launched on May 14, 2018, replacing the paid services of Google Drive to emphasize that the program is used across multiple Google services.

🌐 The W3C published the WebDriver specification as a Recommendation on May 31, 2018, standardising the browser automation protocol that originated in Selenium. Implementations are supported by Google (Chrome), Mozilla (Firefox), and Apple (Safari).

🌐 Hypercorn, a high-performance ASGI and WSGI web server with support for HTTP/1, HTTP/2, and HTTP/3, was first released as open source on June 2, 2018.

🏢 In March 2018, Kubernetes became the first project to graduate from the Cloud Native Computing Foundation (CNCF), signaling its maturity and widespread adoption.

🏢 On June 5, 2018, AWS Elastic Kubernetes Service (EKS) available in the US East (N. Virginia) and US West (Oregon) Regions.

📖 Ryan Dahl, the creator of Node.js, officially announced Deno during his talk "10 Things I Regret About Node.js" at JSConf EU in Berlin on June 6, 2018.

🏢 On June 13, 2018, Microsoft announced the general availability of the Azure Kubernetes Service (AKS).

🏢 Slack acquired the intellectual property of HipChat and Stride from Atlassian on July 26, 2018, subsequently migrating those users to the Slack platform.

☁ Istio 1.0 was released on July 31, 2018, a little over a year after the initial 0.1 release, marking recognition that a core set of functionality had been built that users could rely on for production use.

☁ Knative, an open-source serverless and event-driven application platform built on top of Kubernetes, was released by Google in July 2018, developed in close partnership with VMware, IBM, Red Hat, and SAP.

☁ On August 15, 2018, Google Cloud Run (fully managed) was released in Alpha, providing a serverless execution environment for running stateless request-driven containers built on Knative.

📖 Go 1.11 was released on August 24, 2018, introducing experimental support for modules, a new dependency management system that later became the standard way to manage Go project dependencies.

🔒 TLS 1.3 was defined in RFC 8446 in August 2018.

☁ On September 10, 2018, Microsoft announced another rebranding of VSTS, this time

to "Azure DevOps Services".

⚙️ Podman was initially released in October 2018.

⚙️ Microsoft introduced the Windows Pseudo Console (ConPTY) in 2018 as part of the Windows 10 October 2018 Update, providing a standardized PTY-like interface for the Windows Console to better support terminal-oriented software like OpenSSH and WSL.

🏢 In 2018, Microsoft acquired GitHub, the largest host for open source project infrastructure. The deal closed on October 26, 2018.

☁️ On November 26, 2018, AWS announced Firecracker, an open-source virtualization technology that enables service owners to operate multi-tenant container-based services by combining the speed and resource efficiency of containers with the security and isolation of virtual machines. Firecracker was developed internally at AWS to power AWS Lambda and AWS Fargate, using Micro-Virtual Machines (microVMs) based on Linux KVM and written in Rust.

☁️ On November 28, 2018, announced at AWS re:Invent, Envoy became the third project to graduate from the Cloud Native Computing Foundation, following Kubernetes and Prometheus.

☁️ On 29 Nov 2018, AWS announced Ruby as a supported language for AWS Lambda.

☁️ Kubernetes 1.13, released in December 2018, saw the Container Storage Interface (CSI) and CoreDNS reach general availability.

☁️ GitHub Actions was announced on October 16, 2018, at the GitHub Universe conference as a limited public beta, bringing workflow automation capabilities directly into GitHub repositories.

🌐 FastAPI is a modern, high-performance web framework for building APIs with Python, created by Sebastián Ramírez and first released as open source on December 24, 2018.

💻 Visual Studio Code reached the top spot in the Stack Overflow Developer Survey in 2018, becoming the most popular development environment among respondents.

💻 Tree-sitter, an incremental parsing library and parser generator tool developed by Max Brunsfeld, was originally developed by GitHub for use in the Atom text editor, where it was first released in 2018. It builds concrete syntax trees for source code and is used for syntax highlighting, code navigation, and static analysis.

2019

⚙️ In January 2019, Linuxbrew was merged back into Homebrew, adding beta support for Linux and the Windows Subsystem for Linux to the Homebrew feature set.

📊 Trino is an open-source distributed SQL query engine designed to query large data sets distributed over one or more heterogeneous data sources. In January 2019, the original creators of Presto, Martin Traverso, Dain Sundstrom, and David Phillips, created a fork of the Presto project. They initially kept the name Presto and used the PrestoSQL web handle to distinguish it from the original PrestoDB project.

🧠 Generative Pre-trained Transformer 2 (GPT-2) is an open-source artificial intelligence created by OpenAI in February 2019. GPT-2 translates text, answers questions, summarizes passages, and generates text output on a level that, while sometimes indistinguishable from that of humans, can become repetitive or nonsensical when generating long passages.

☁️ On April 9, 2019, Google announced Cloud Run in Beta at Google Cloud Next 2019, introducing a fully managed serverless platform that runs stateless containers using the Knative serving API.

💻 The Visual Studio Code Remote Development extension pack, enabling developers to use a container, remote machine, or the Windows Subsystem for Linux (WSL) as a full-featured development environment, was first announced on May 2, 2019.

⚙️ Microsoft announced WSL 2 on May 6, 2019, and it was shipped with Windows 10 version 2004.

📱 On 7 May 2019, Google announced that the Kotlin programming language is now its preferred language for Android app developers.

☁️ Argo CD 1.0 was released on May 16, 2019, marking the tool's first stable release for automating the deployment of desired application states in specified target environments.

⚙️ At WWDC on June 3, 2019, Apple announced that the variant of iOS running on iPad would be rebranded as "iPadOS," signaling the company's intention to develop the iPad platform in a direction distinct from the iPhone.

🌐 Tailwind CSS is an open-source CSS framework. Unlike other frameworks, like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables. Instead, it creates a list of "utility" CSS classes that can be used to style each element by mixing and matching. It was initially released on 13 May 2019.

💻 cURL 7.66, released 11 September 2019, supports HTTP/3 (and thus QUIC).

⚙️ iPadOS 13.1 was released on September 24, 2019, as the first version of the newly branded iPad operating system, featuring an updated Home Screen with widgets, improved multitasking with Split View and Slide Over, desktop-class Safari browsing, and mouse and trackpad support.

💻 macOS Catalina, released on October 7, 2019, adopted Zsh as the default login shell, replacing the GPLv2-licensed version of Bash that had been the default since Mac OS X Panther in 2003. Apple announced the change at WWDC on June 3, 2019, and shipped Zsh 5.7.1, the most current version at the time.

☁️ GitHub Actions became generally available on November 13, 2019, offering a community-led approach to software automation with full CI/CD capabilities for DevOps and native package management.

☁️ On November 14, 2019, Google Cloud Run (fully managed) became Generally Available (GA), offering a Knative-based serverless compute platform with a 99.95% monthly uptime SLA.

⚙️ On November 25, 2019, the RIPE NCC announced that it had fully run out of IPv4 addresses, underscoring the continued need to migrate to IPv6.

🌐 WebAssembly became a World Wide Web Consortium recommendation on 5 December 2019.

📖 In 2019, the JS Foundation and Node.js Foundation merged to form the OpenJS Foundation.

📖 Rust added support for `async/await` with version 1.39.0 in 2019.

📖 The 20th Anniversary Edition of *The Pragmatic Programmer* was released in 2019, introducing the "ETC" (Easier To Change) principle as the foundational meta-principle of software design, which asserts that the essence of all good design is making it easier to change.

Timeline - 2020-24

2020

🌐 Playwright, a framework for reliable end-to-end testing for modern web apps, was first released as open source by Microsoft on January 31, 2020. It was created by engineers who had previously worked on Puppeteer at Google, extending cross-browser automation to Chromium, Firefox, and WebKit under a single API.

💻 PowerShell 7.0 was released on March 4, 2020, dropping the "Core" suffix and

serving as the successor to both Windows PowerShell 5.1 and PowerShell Core 6.x with improved Windows module compatibility.

☁ Istio 1.5 was released on March 5, 2020, dramatically simplifying the experience of installing, running, and upgrading Istio by "embracing the monolith" and consolidating the control plane into a single new binary, Istiod.

📖 Hyrum's law, an observation that "with a sufficient number of users of an API, it does not matter what you promise in the contract: all observable behaviors of your system will be depended on by somebody," was articulated by Google engineer Hyrum Wright and named by his colleague Titus Winters in the book *Software Engineering at Google*, published in March 2020.

☁ On April 7, 2020, the CNCF Technical Oversight Committee voted to accept Argo as an incubation-level hosted project, a set of Kubernetes-native tools for running and managing jobs and applications on Kubernetes.

🌐 Vite, a frontend build tool created by Evan You, was first announced on April 20, 2020, leveraging native ES Modules to eliminate bundling during development and achieve near-instant hot module replacement. Vite 2.0, redesigned to be framework-agnostic, was released in February 2021.

💻 GitHub Codespaces was announced on May 6, 2020, as a limited public beta at GitHub Satellite 2020, providing fully-featured cloud-hosted development environments directly within GitHub powered by Visual Studio Code.

🔒 GitHub Advanced Security was announced in May 2020 at GitHub Satellite 2020, introducing code scanning in beta and secret scanning for private repositories in limited public beta to help automate security directly into the developer workflow.

📖 Deno v1.0.0, a secure-by-default runtime for JavaScript and TypeScript with a single-executable design, was officially released on May 13, 2020.

🌐 In May 2020, Discord transitioned its primary domain from discordapp.com to discord.com, signaling its broader goal of becoming a place for all communities, not just gamers.

🧠 Generative Pre-trained Transformer 3 (GPT-3) is an autoregressive language model released in June 2020 that uses deep learning to produce human-like text. Given an initial text as prompt, it will produce text that continues the prompt.

⚙️ Apple announced its plan to transition the Mac from Intel processors to its own Apple Silicon at WWDC on June 22, 2020, marking the third major processor architecture transition in Mac history.

 The Rust for Linux project was announced in July 2020 in the Linux kernel mailing list with goals of leveraging Rust's memory safety to reduce bugs when writing kernel drivers.

 Visual Studio Codespaces consolidated into GitHub Codespaces on September 4, 2020, with Microsoft announcing that the two distinct experiences were causing confusion and that the Azure-based Visual Studio Codespaces portal would be retired on February 17, 2021.

 GitHub code scanning reached general availability in September 2020, four months after the beta release, having been used to scan more than 12,000 repositories and finding over 20,000 security issues including remote code execution, SQL injection, and cross-site scripting flaws.

 On September 21, 2020, Homebrew version 2.5.2 was released with support for bottle taps (binary package repositories) via GitHub Releases.

 On October 6, 2020, Google rebranded G Suite as Google Workspace, introducing a new integrated user experience and unified brand identity for its communication and collaboration apps.

 Jakarta Server Faces 3.0.0 was released on October 28, 2020, with the package name changed from javax to jakarta as part of the Jakarta EE transition

 The first Apple Silicon Macs — the MacBook Air, 13-inch MacBook Pro, and Mac mini with the M1 chip — were announced on November 10, 2020, and began shipping on November 17, 2020, delivering significant performance and energy efficiency improvements over their Intel predecessors.

 Microsoft released .NET 5.0 on November 10, 2020. The "Core" branding was removed and version 4.0 was skipped to avoid conflation with .NET Framework, which remains the Windows-specific product. It addresses the patent concerns related to the .NET Framework.

 macOS 11 Big Sur was released on November 12, 2020, the first version to increment the major version number beyond 10 since the original Mac OS X release in 2001. It was also the first macOS release to support Apple Silicon (M1) Macs, featuring a redesigned user interface inspired by iOS.

 The most recent major update to "The Scrum Guide" was published by Ken Schwaber and Jeff Sutherland on November 18, 2020.

 Ruby 3.0.0 was released on Christmas Day in 2020.

 In December 2020, PrestoSQL was rebranded as Trino.

 In 2020, with the introduction of the LAMBDA function, Microsoft Excel became a Turing-complete programming language, allowing users to define new functions written in Excel's own formula language.

 Tomcat 10.0, released in 2020, implemented Servlet 5.0 and JSP 3.0, and was the first version to migrate from the `javax.` to the `jakarta.` package namespace following the transfer of Java EE to the Eclipse Foundation as Jakarta EE.

2021

 In January 2021, Elastic announced it was relicensing Elasticsearch and Kibana from the Apache License 2.0 to dual licensing under the Server Side Public License and the Elastic License, in response to Amazon Web Services offering Elasticsearch and Kibana as a service without adequate collaboration.

 On January 21, 2021, Amazon Web Services announced a fork of Elasticsearch and Kibana under the Apache License 2.0, which it rebranded as OpenSearch in April 2021 due to trademark conflicts.

 On February 8, 2021, the formation of the Rust Foundation was announced by its five founding companies (AWS, Huawei, Google, Microsoft, and Mozilla).

 Google Kubernetes Engine (GKE) Autopilot was introduced in February 2021, providing a managed mode for clusters.

 The IEEE 802.11ax-2021 standard (Wi-Fi 6) was approved on February 9, 2021.

 In March 2021, Lucene changed its logo, and Apache Solr became a top-level Apache project again, independent from Lucene.

 Secret scanning for private repositories became generally available on March 31, 2021, for all GitHub Advanced Security customers on GitHub Enterprise Cloud, with expanded pattern coverage from 24 to 37 partners and having helped organizations find and revoke more than 5,000 secrets during the beta period.

 In May 2021, the IETF standardized QUIC in RFC 9000.

 QUIC support in Firefox arrived in May 2021.

 Vertex AI, a unified, managed machine learning platform, was announced at Google I/O in May 2021.

 NocoDB, an open-source no-code platform that turns any database into a smart spreadsheet, was officially launched on May 26, 2021.

☁ Terraform 1.0 was released on June 08, 2021.

☁ Deno Deploy, a distributed HTTP service that allows users to run JavaScript, TypeScript, and WebAssembly at the edge, was launched on June 23, 2021.

🧠 GitHub Copilot is a cloud-based artificial intelligence tool developed by GitHub and OpenAI to assist users of Visual Studio Code, Visual Studio, Neovim, and JetBrains integrated development environments (IDEs) by autocompleting code, announced for technical preview on June 29, 2021.

💻 With the 0.5 release of Neovim on July 2, 2021, it gained built-in support for the Language Server Protocol (LSP), Tree-sitter, and more complete Lua support.

🏢 Salesforce officially completed its acquisition of Slack for approximately \$27.7 billion on July 21, 2021, marking one of the largest acquisitions in the software industry.

🏢 Microsoft discontinued the hosted Skype for Business Online service on July 31, 2021, fully transitioning its communication and collaboration capabilities to Microsoft Teams.

🧠 OpenAI Codex is an artificial intelligence model developed by OpenAI that parses natural language and generates code in response, serving as the foundation for GitHub Copilot. As a descendant of OpenAI's GPT-3 model fine-tuned for programming applications, Codex was released on August 10, 2021.

💻 GitHub Codespaces, a cloud-hosted development environment powered by Visual Studio Code, reached general availability for GitHub Team and Enterprise Cloud customers on August 11, 2021.

🏢 The eBPF Foundation was launched under the Linux Foundation in August 2021. Founding members included Google, Facebook, Isovalent, Microsoft, and Netflix, reflecting eBPF's growing importance in the industry.

🔒 OpenSSL 3.0 was released on September 7, 2021.

⚙️ Windows 11 was officially released on October 5, 2021, featuring a major visual redesign with a centered Taskbar and Start menu, improved window snapping ("Snap Layouts"), and a focus on enhanced security requirements.

💻 On October 27, 2021, GitHub released the GitHub Copilot Neovim plugin as a public repository.

☁ GitHub Actions added support for OpenID Connect (OIDC) on October 27, 2021, enabling secure cloud deployments with short-lived tokens that are automatically rotated for each deployment.

🐛 Log4Shell was a zero-day vulnerability in Log4j, a popular Java logging framework, involving arbitrary code execution that had existed unnoticed since 2013 and was privately disclosed to the Apache Software Foundation on November 24, 2021 by Chen Zhaojun of Alibaba Cloud's security team.

☁️ GitHub Actions reusable workflows became generally available on November 24, 2021, allowing users to reuse entire workflows as if they were actions and reducing duplication.

☁️ Azure Container Apps, a fully managed serverless container service built on Kubernetes, was introduced in public preview on November 2, 2021.

☁️ Knative reached version 1.0 in November 2021, with all its repositories designated by the community as stable and suitable for commercial use.

🏠 Grafana acquired k6 in 2021.

🌐 Ruby on Rails 7.0 was released on December 15, 2021, replacing Node.js and Webpack with import maps by default and introducing the Hotwire family of libraries.

📖 Vitest was first released in December 2021 as a Vite-native unit testing framework for JavaScript, reusing Vite's transform pipeline and configuration to deliver near-instant test startup for Vite-based projects. Vitest 1.0 reached production-ready status in December 2023.

🧠 LaMDA (Language Model for Dialogue Applications) is a family of conversational large language models developed by Google, originally introduced as Meena in 2020 and evolved into the first-generation LaMDA announced at the 2021 Google I/O keynote, followed by the second generation the following year.

2022

☁️ On March 2, 2022, Knative was accepted as a Cloud Native Computing Foundation (CNCF) incubating project, after Google submitted it for consideration in November 2021.

📖 Go 1.18 was released on March 15, 2022, adding support for generic programming (type parameters), the most significant language change since Go 1.0 and a long-requested community feature.

⚙️ Chocolatey 1.0.0 was officially released on March 18, 2022, marking a major milestone in version stability and enterprise readiness after over a decade of development.

🧠 PaLM (Pathways Language Model) is a 540 billion parameter transformer-based

large language model developed by Google AI, first announced in April 2022 and remaining private until Google launched a public API in March 2023.

🧠 On May 11, 2022, Google unveiled LaMDA 2, the successor to LaMDA, during the 2022 Google I/O keynote.

🏠 Meta transferred Jest to the OpenJS Foundation in May 2022, reflecting the framework's growth beyond Meta's internal use into a community-maintained project hosted alongside Node.js, jQuery, and webpack.

☁️ Kubernetes 1.24, released in May 2022, removed "dockershim," officially ending built-in support for Docker as a container runtime in favor of the Container Runtime Interface (CRI).

☁️ Azure Container Apps, a fully managed serverless container service built on Kubernetes, reached general availability on May 24, 2022.

🧠 On June 21, 2022, GitHub announced that Copilot was out of "technical preview" and made available as a subscription-based service for individual developers.

🌐 Fresh 1.0, a full-stack web framework for Deno that uses an "islands" architecture for zero-config and high performance, was released on June 28, 2022.

🌐 Bun, a fast all-in-one JavaScript runtime and toolkit written in Zig and powered by JavaScriptCore, entered public beta with the release of version 0.1.0 on July 5, 2022.

☁️ On August 25, 2022, Heroku announced plans to eliminate its free tier plans, citing fraud and abuse as the primary reasons, with the removal scheduled for November 28, 2022.

⚙️ Systemd support became available in WSL on September 21, 2022.

☁️ Istio was accepted to the Cloud Native Computing Foundation on September 30, 2022 at the incubating maturity level.

🧠 ReAct (Synergizing Reasoning and Acting in Language Models), a general paradigm that combines reasoning and acting with Large Language Models (LLMs) to perform tasks, was first submitted to arXiv by researchers from Princeton University and Google Research on October 6, 2022.

🏠 On October 13, 2022, Microsoft announced it would phase out the "Microsoft Office" brand in favor of "Microsoft 365," subsequently rebranding Office.com and the Office mobile apps to reflect this shift.

⚙️ Linux 6.0 was released on October 2, 2022, focusing on performance improvements

for modern CPU architectures and including the initial groundwork for Rust support in the kernel.

 In October 2022, a pull request for accepting the implementation for Rust for Linux was approved by Torvalds.

 LangChain, an open-source framework designed to simplify the creation of applications using large language models (LLMs), was launched by Harrison Chase in October 2022.

 GitHub Codespaces was made available to all individual users on November 9, 2022, including a free monthly usage quota for the free tier.

 On December 6, 2022, the Cloud Native Computing Foundation announced that Argo had graduated, joining the ranks of Kubernetes, Prometheus, and Envoy as a top-level project.

 OCaml 5.0 was released on December 16, 2022, featuring a complete rewrite of the runtime to support shared-memory parallelism and effect handlers for lock-free concurrency.

 Linux 6.1 was released in December 2022, including the ability to write eBPF programs in Rust, further expanding the ecosystem for safe and performant kernel-level observability and networking.

 ChatGPT (Chat Generative Pre-trained Transformer) is an artificial intelligence chatbot developed by OpenAI and launched in November 2022, built on top of OpenAI's GPT-3.5 and GPT-4 families of large language models and fine-tuned using both supervised and reinforcement learning techniques.

 Support for npm packages via the `npm:` specifier became stable in Deno v1.28 on November 14, 2022, allowing the runtime to tap into the vast existing JavaScript ecosystem.

 HTTP/3, the successor to HTTP/2, was published in 2022.

2023

 On February 7, 2023, Microsoft announced a major overhaul to Bing including the addition of chatbot functionality marketed as "the new Bing".

 Azure OpenAI Service reached general availability on January 17, 2023, providing enterprise access to OpenAI's large language models.

 Llama (acronym for Large Language Model Meta AI, and formerly stylized as LLaMA)

is a family of autoregressive large language models (LLMs) released by Meta AI starting in February 2023.

🧠 Notion AI, an integrated AI assistant for the Notion workspace, reached general availability on February 22, 2023, after a private alpha period that began in November 2022.

🧠 Generative Pre-trained Transformer 4 (GPT-4) is a multimodal large language model created by OpenAI as the fourth in its series of GPT foundation models, launched on March 14, 2023 and made publicly available through the paid ChatGPT Plus chatbot, OpenAI's API, and the free Microsoft Copilot chatbot.

🧠 BabyAGI, a task-driven autonomous agent created by Yohei Nakajima, was publicly shared on Twitter on March 28, 2023. Written as a 105-line Python script using GPT-4, Pinecone, and LangChain, it demonstrated autonomous task planning and execution, going viral and accumulating over 16,000 GitHub stars.

🧠 AutoGPT, an open-source autonomous AI agent created by Toran Bruce Richards of Significant Gravititas Ltd., was released on GitHub on March 30, 2023. One of the first applications to showcase GPT-4's autonomous capabilities, it broke down user-defined goals into sub-tasks using tools like web browsing, file management, and API calls, becoming the fastest-growing open-source project in GitHub history with over 100,000 stars within weeks.

🧠 Amazon Bedrock was first announced on April 13, 2023, in a limited preview as a fully managed service that makes foundation models (FMs) from leading AI startups and Amazon available via an API.

🧠 Bard is a conversational artificial intelligence chatbot developed by Google, based on the LaMDA family of large language models. It was developed as a response to the rise of OpenAI's ChatGPT, and was released in a limited capacity in March 2023 to lukewarm responses.

🧠 Claude 1, the original large language model developed by Anthropic, was released in March 2023, establishing the foundation for Anthropic's family of conversational AI models.

⚙️ Chocolatey 2.0.0 was released on May 31, 2023, featuring a major update to the underlying NuGet client libraries and introducing support for Semantic Versioning (SemVer) 2.0.0.

☁️ Istio moved to the graduated maturity level within the Cloud Native Computing Foundation on July 12, 2023, less than a year after entering the incubator.

🔐 Azure Active Directory (Azure AD) was officially renamed to Microsoft Entra ID on

July 15, 2023.

🧠 MetaGPT, an innovative meta-programming framework that incorporates efficient human workflows into LLM-based multi-agent collaborations, was introduced in a paper published on August 1, 2023. It encodes Standardized Operating Procedures (SOPs) into prompt sequences and uses an assembly line paradigm to assign diverse roles to agents, breaking down complex tasks into subtasks that many agents work on together.

📖 Go 1.21 was released on August 8, 2023, introducing built-in min, max, and clear functions, a new log/slog structured logging package, and formal toolchain management.

☁️ HashiCorp relicensed Terraform and its other products from the Mozilla Public License v2.0 to the Business Source License (BUSL) v1.1 on August 10, 2023, a source-available license that prohibits competitive use, triggering the community response that produced the OpenTF manifesto.

☁️ Dagou, a local-first and DAG-based workflow engine designed as a "zero-ops" alternative to more complex orchestrators like Apache Airflow, was first released as open source on GitHub on August 11, 2023.

☁️ Following HashiCorp's announcement to move Terraform to a non-open-source license, the OpenTF project (later rebranded as OpenTofu) was officially forked on August 25, 2023. It joined the Linux Foundation on September 20, 2023.

🌐 Bun 1.0, marking the transition of the runtime to a stable, production-ready version with an integrated bundler and test runner, was officially released on September 8, 2023.

🧠 Amazon Bedrock reached general availability (GA) on September 28, 2023, launching with support for foundation models from AI21 Labs, Anthropic, Cohere, Stability AI, and Amazon.

🧠 In November 2023, GitHub Copilot Chat was updated to use OpenAI's GPT-4 model.

🧠 Google Gemini is a family of multimodal large language models developed by Google DeepMind as the successor to LaMDA and PaLM 2, comprising Gemini Ultra, Gemini Pro, Gemini Flash, and Gemini Nano. It was announced on December 6, 2023 as a direct competitor to OpenAI's GPT-4.

📖 Mojo is a programming language in the Python family currently under development, available in browsers via Jupyter notebooks and locally on Linux and macOS, first appearing in 2023.

🧠 LocalAI is a free, open-source OpenAI alternative that acts as a drop-in replacement REST API compatible with OpenAI API specifications for local inferencing, first appearing in 2023.

💻 Cursor, an AI-native fork of Visual Studio Code (VS Code) developed by Anysphere, was publicly launched in 2023.

2024

🧠 In February 2024, Google launched Gemini 1.5 in a limited capacity, positioned as a more powerful and capable model than 1.0 Ultra.

🧠 LangGraph, a library built on top of LangChain to enable the creation of stateful, multi-actor applications with LLMs by creating cyclic graphs, was announced on January 17, 2024.

☁ OpenTofu reached general availability (GA) with its 1.6.0 release on January 10, 2024, marking its first stable production-ready version.

🧠 On February 8, 2024, Google introduced the AI Premium plan for Google One, which provides access to Gemini Advanced and Gemini in Gmail, Docs, Slides, Sheets, and Meet.

🧠 On February 27, 2024, GitHub Copilot Enterprise reached general availability at \$39 per month, extending the Business plan with the ability to reference an organization's internal codebase and knowledge base.

🧠 Devin, branded as the "world's first fully autonomous AI software engineer," was officially released by Cognition AI on March 12, 2024.

☁ Envoy Gateway 1.0 was released on March 13, 2024, reaching general availability as a performant and functional implementation of the Kubernetes Gateway API.

🔒 GitHub announced Copilot Autofix for GitHub Advanced Security in public beta on March 20, 2024, providing AI-powered remediation that analyzes vulnerabilities in code and offers suggestions to help developers fix them.

🧠 The Claude 3 family of large language models was announced by Anthropic on March 4, 2024, introducing three model tiers: Claude 3 Opus (maximum capability), Claude 3 Sonnet (balanced performance), and Claude 3 Haiku (low latency and cost). The Claude 3 family marked Anthropic's first multimodal models, adding vision capabilities (image understanding) to all three tiers. Opus and Sonnet were made generally available on March 4, 2024, while Haiku became generally available on March 13, 2024.

🌐 Bun 1.1, which introduced stable support for Windows and significant performance improvements for common operations, was released on April 1, 2024.

🧠 On April 29, 2024, GitHub introduced Copilot Workspace as a technical preview, a Copilot-native developer environment that turns a natural-language GitHub issue into a spec, a plan, and code changes.

🧠 GPT-4o (GPT-4 Omni) is a multilingual, multimodal generative pre-trained transformer designed by OpenAI, announced by OpenAI's CTO Mira Murati during a live-streamed demonstration on May 13, 2024 and released the same day.

🧠 Firebase Genkit, an open-source framework by Google for building and deploying AI-powered applications, was first introduced during the Google I/O 2024 keynote on May 14, 2024.

🔧 Copilot Autofix for CodeQL code scanning alerts reached general availability on August 14, 2024, enabling developers to fix code vulnerabilities more than three times faster with a median time of 28 minutes compared to 1.5 hours manually.

📊 On August 29, 2024, Elastic added the GNU Affero General Public License as a third licensing option for Elasticsearch, declaring the software "Open Source, Again."

✂ Unicode 16.0 was released on September 10, 2024, adding 5,185 new characters including seven new scripts and seven new emoji, for a total of 154,998 characters.

🌐 Deno v2.0.0 was officially released on October 9, 2024, focusing on full backward compatibility with Node.js and npm alongside long-term support (LTS) for enterprise use.

🧠 On September 16, 2024, Microsoft announced "Microsoft 365 Copilot Wave 2," introducing "Copilot agents"—specialized AI assistants that can be built in Copilot Studio or directly within SharePoint. These agents entered general availability on October 17, 2024.

📊 PostgreSQL 17 was released on September 26, 2024, providing significant performance boosts to vacuuming and I/O, along with expanded JSON functionality.

⚙ The IEEE 802.11be-2024 amendment (Wi-Fi 7) was approved by the IEEE SA Standards Board on September 26, 2024.

🧠 BabyAGI v2, a major rewrite of the original BabyAGI framework by Yohei Nakajima, was released in September 2024. It introduced the functionz framework for storing functions and metadata in a database, reflecting advancements in GPT-4o, Claude tool use, and structured outputs since the original release.

🧠 Anthropic introduced the Computer Use capability in public beta for Claude 3.5 Sonnet on October 22, 2024, enabling the model to navigate desktop environments by interpreting screenshots and simulating mouse and keyboard inputs.

🧠 The Model Context Protocol (MCP) is an open standard, open-source framework introduced by Anthropic on November 25, 2024 to standardize the way artificial intelligence systems like large language models integrate and share data with external tools, systems, and data sources.

🧠 OpenAI o3 is a reflective generative pre-trained transformer model developed by OpenAI as a successor to OpenAI o1, designed to devote additional deliberation time for questions requiring step-by-step logical reasoning, announced on December 20, 2024.

🧠 Devin became generally available in December 2024, offered to engineering teams starting at \$500 per month with no seat limits, alongside a Slack integration, an IDE extension, and an API.

⚙️ In 2024, the Single UNIX Specification Issue 8 adopted more inclusive terminology for pseudoterminals, officially replacing the traditional "master" and "slave" labels with "manager" and "subsidiary" respectively.

📖 Elixir v1.18 was released on December 19, 2024, featuring built-in JSON support and the first iteration of type-checking for calls.

Timeline - 2025-Present

2025

🧠 A significant API update for Anthropic's Computer Use was released on January 24, 2025 (header: `anthropic-beta: computer-use-2025-01-24`), improving the model's accuracy and reliability in navigating complex UI environments.

🧠 On January 30, 2025, Google released Gemini 2.0 Flash as the new default model, with Gemini 1.5 Flash still available for usage. This was followed by the release of Gemini 2.0 Pro on February 5, 2025.

🧠 GitHub officially introduced Copilot Agent Mode on February 6, 2025, as part of a major announcement highlighting new autonomous workflows and productivity enhancements for developers. Subsequently, on February 24, 2025, GitHub released a follow-up blog post confirming availability in VS Code Stable/Insiders, marking its first broader rollout beyond early access.

💻 fish 4.0.0 was released on February 27, 2025, rewritten in Rust from the previous

C++ implementation, marking the largest architectural change in the shell's history.

🧠 In February 2025, Amazon introduced Alexa+, the latest version of its voice assistant powered by generative AI and free for all Prime members; the same announcement revealed that Anthropic's Claude models were incorporated into Alexa+.

🧠 Devin 2.0 was released on April 3, 2025, introducing a new agent-native IDE experience that lets users run multiple parallel Devins, while cutting the starting price from \$500 to \$20 per month.

🔐 GitHub unbundled GitHub Advanced Security on April 1, 2025, making it available as two standalone products: GitHub Secret Protection (\$19/month per active committer) and GitHub Code Security (\$30/month per active committer), broadening access to GitHub Team plan customers.

☁ Heroku Fir, the next generation of the Heroku platform built on cloud-native technologies including Cloud Native Buildpacks and native OpenTelemetry, became generally available on April 14, 2025.

🧠 GitHub officially announced the rollout of the Agent Mode to all users on April 4, 2025, which included support for the Model Context Protocol (MCP) and the introduction of the Copilot code review agent.

☁ On May 6, 2025, Grafana k6 v1.0.0 was released after 9 years of iteration and countless community contributions.

🧠 At Config 2025 on May 7, 2025, Figma unveiled a suite of AI-powered tools including Figma Sites, Figma Buzz, Figma Draw, and Figma Make, an AI tool that turns prompts into working apps and is powered by Anthropic's Claude 3.7 Sonnet model.

🧠 On May 17, 2025, GitHub announced the GitHub Copilot coding agent, a more autonomous mode of operation that can be assigned issues and open pull requests on a developer's behalf.

🧠 At Microsoft Build 2025 on May 19, 2025, GitHub announced that GitHub Copilot now includes an asynchronous coding agent, embedded directly in GitHub and accessible from VS Code, which excels at low-to-medium complexity tasks by spinning up secure development environments powered by GitHub Actions.

💻 Claude Code, Anthropic's AI-powered command-line coding assistant, was first introduced as a beta research preview alongside Claude 3.7 Sonnet on February 24, 2025. It then became generally available on May 22, 2025.

🧠 The release of Claude 4 (Opus and Sonnet) on May 22, 2025, further integrated computer use capabilities into the model's core agentic workflows, featuring improved

pixel-counting and reduced latency.

🧠 On June 17, 2025, Google announced general availability for 2.5 Pro and Flash. They also introduced Gemini 2.5 Flash-Lite that same day, a model optimized for speed and cost-efficiency.

💻 Gemini CLI was officially launched by Google on June 25, 2025, as an open-source command-line interface for Gemini 2.5 Pro, enabling AI-powered coding, debugging, research, and more directly in terminal environments.

📖 Jakarta EE 11, released on June 26, 2025, modernized the platform around Java SE 21 and 17, adding the Jakarta Data specification for streamlined data access.

🧠 The GitHub Copilot Chat extension, which powers the agent's capabilities, was released as open-source software under the MIT License on June 30, 2025.

🧠 General availability of the GitHub Copilot coding agent and MCP support was achieved in the stable Visual Studio Code June release (v1.102) on July 9, 2025.

🧠 On August 6, 2025, Jules was officially out of beta and launched publicly, powered by Gemini 2.5.

🧠 GPT-5 is a multimodal large language model developed by OpenAI and the fifth in its series of generative pre-trained transformer (GPT) foundation models. Preceded in the series by GPT-4, it was launched on August 7, 2025, combining reasoning capabilities and non-reasoning functionality under a common interface.

🧠 The August update (v1.104) of Visual Studio Code, released on September 12, 2025, added "Auto" model selection and introduced support for **AGENTS.md** files to provide custom instructions for the GitHub Copilot Agent.

🐛 The Shai-Hulud npm supply chain attack, a self-propagating worm that harvested sensitive data and automatically published malicious versions of accessible packages, was first detected on September 15, 2025.

🧠 Notion 3.0 was launched on September 18, 2025, rebranding its AI capabilities as "Notion AI Agents" and introducing autonomous, multi-step agentic workflows within the workspace.

📊 PostgreSQL 18 was released on September 25, 2025, introducing asynchronous I/O (AIO) to significantly enhance the performance of concurrent I/O tasks.

💻 GitHub Copilot CLI was announced as entering public preview on September 25, 2025, bringing the Copilot coding agent directly to the terminal with terminal-native development, GitHub integration, agentic capabilities, MCP-powered extensibility, and

user control features.

🧠 On September 29, 2025, Anthropic enhanced the tool with Claude Code v2.0, introducing checkpoints, a native VS Code extension, terminal UX improvements, and the Claude Agent SDK.

🧠 Anthropic's Skills API ([anthropic-beta: skills-2025-10-02](#)) was officially introduced on October 2, 2025, providing a standardized way to package workflow instructions, executable code, and documentation into reusable capabilities for Claude via [SKILL.md](#) files.

☁️ On October 8, 2025, the Cloud Native Computing Foundation announced the graduation of Knative, a serverless, event-driven application layer on top of Kubernetes, marking its readiness for widespread production use.

🧠 On October 15, 2025, Anthropic launched the Skills Registry, a centralized platform for discovering and sharing "Anthropic-verified" and community-contributed skills to enhance Claude's functionality.

🌐 A global DNS misconfiguration on October 29, 2025, disrupted Microsoft 365, Xbox Live, and major Azure corporate clients.

🧠 LangChain and LangGraph officially reached their Version 1.0 milestones on October 22, 2025. This release marked LangGraph's transition to a "durable agent framework" with a commitment to API stability, featuring built-in support for persistent state, human-in-the-loop (HITL) workflows, and a redesigned documentation site.

💻 Cursor 2.0 was released on October 29, 2025, introducing "Composer"—an AI-native coding interface optimized for orchestrating multi-agent workflows and high-speed code generation across codebases.

☁️ On November 14, 2025, AWS Lambda added support for building serverless applications using Rust.

🧠 Google officially launched Gemini 3 on November 18, 2025 — a major release that brought enhanced reasoning, multimodal understanding, and agentic capabilities to the Gemini app, AI Search, Google AI Studio, Vertex AI, and more.

🧠 Google Antigravity platform was announced on November 18, 2025, alongside the release of Gemini 3, representing Google's entry into AI-powered development tools as an AI agent-first IDE heavily modified from Visual Studio Code.

🧠 Support for the Model Context Protocol (MCP) within the Anthropic Skills framework was achieved on November 20, 2025, allowing skills to seamlessly interact with external tools and data sources via standardized MCP servers.

🌐 Anthropic announced the acquisition of Bun on December 2, 2025, with the aim of using the runtime to power its agentic workflows and developer tools.

🏢 The Agentic AI Foundation (AAIF) was founded on December 9, 2025 as a new foundation under the Linux Foundation umbrella to provide neutral governance for the rapidly evolving ecosystem of AI agents.

🧠 GPT-5.2, an incremental update to GPT-5, was released on December 11, 2025, featuring improved reasoning capabilities, reduced latency, and enhanced tool integration. This release maintained backward compatibility while introducing more specialized modes for different use cases.

📖 Ruby 4.0.0 was released on Christmas Day in 2025.

💻 In 2025, Neovim was voted the most admired development environment in the Stack Overflow developer survey for the fifth consecutive year.

2026

🧠 Mastra 1.0, an all-in-one open-source TypeScript framework for building, iterating, and deploying AI agents, was released on January 20, 2026. Developed by the team behind Gatsby, it provides integrated tools for workflows, RAG, memory, and observability.

🧠 Support for Anthropic's Claude models within the GitHub Copilot Agent and enhanced multi-agent session management were introduced in the January 2026 release (v1.109) of Visual Studio Code.

🧠 GPT-5.3-Codex, a large language model by OpenAI focused on code generation, speed, and the ability to search repositories, run terminal commands, and debug code, was released on February 5, 2026. Positioned against Anthropic's Claude Opus 4.6 and reported to be 25% faster on technical benchmarks, it was initially available only to ChatGPT Pro subscribers.

☁️ On February 6, 2026, Salesforce moved Heroku into sustaining engineering mode, ceasing new feature development and limiting the platform to maintenance, security patches, and stability fixes only.

🧠 On February 17, 2026, Figma partnered with Anthropic to launch Code to Canvas, a feature that converts code generated by AI tools like Claude Code into fully editable, native Figma layers, components, and auto-layout groups.

🧠 The February 2026 release (v1.110) of Visual Studio Code introduced "Agent Plugins," experimental "Agentic Browser Tools," and the ability for agents to share

memory across different sessions for GitHub Copilot.

 GitHub Copilot CLI became generally available for all Copilot subscribers on February 25, 2026, after five months of public preview during which hundreds of improvements were shipped based on user feedback.

 On March 9, 2026, Microsoft announced "Microsoft 365 Copilot Wave 3," introducing "Copilot Cowork"—an autonomous agentic workflow system—alongside "Agent 365," a dedicated control plane for enterprise-wide agent management and governance.

 The existence of a model named Claude Mythos became publicly known on March 26, 2026, due to leaked blog post drafts.

 Microsoft Agent Framework 1.0, a modern AI orchestration framework designed for building autonomous agentic workflows, was first released as an open-source project on April 3, 2026.

 On April 7, 2026, Anthropic announced Project Glasswing, releasing Mythos Preview to 11 companies.

 Linux 7.0 was released on April 12, 2026, as the latest major stable version, continuing the kernel's evolution with advanced hardware support and further integration of memory-safe programming languages.

 Anthropic launched Claude Design on April 17, 2026, a new Anthropic Labs product powered by Claude Opus 4.7 that lets users collaborate with Claude to create polished visual work like designs, prototypes, slides, and one-pagers, available in research preview on the Pro, Max, Team, and Enterprise plans.

 GPT-5.5, a major update to OpenAI's GPT-5 series, was released on April 23, 2026, representing a significant leap in capabilities with improved reasoning, multimodal understanding, and agentic features. API availability was announced on April 24, 2026, bringing the latest model to developers through OpenAI's API endpoints.

 Google announced Antigravity 2.0 at Google I/O on May 19, 2026, introducing Antigravity CLI as a Go-based terminal agent that replaces Gemini CLI, providing a lightweight interface to the same agent runtime used by the desktop app.

 On June 2, 2026, Anthropic expanded access to Claude Mythos for cyber-security, making it available to 150 organisations.

 On June 4, 2026, Anthropic published "When AI builds itself," an Anthropic Institute paper stating that more than 80% of the code merged into its own codebase was now authored by Claude, up from low single digits when Claude Code launched in February 2025, and mapping a path toward "recursive self-improvement," a process in which AI

systems could autonomously design, build, and train their own successors; the paper called for a verifiable global pause mechanism to let alignment research and societal structures keep pace with the technology.

🧠 On June 9, 2026, Anthropic released Claude Fable 5, a Mythos-class model that it had made safe for general use, alongside the restricted Claude Mythos 5; the model shipped with a 1 million token context window and up to 128,000 output tokens per request.

🧠 On June 12, 2026, Anthropic disabled all access to its Mythos-class models, Claude Fable 5 and Claude Mythos 5, to comply with a directive from the United States Department of Commerce to suspend access by foreign nationals, after a jailbreak in Fable 5 was flagged to the White House.

☁ On June 22, 2026, AWS introduced Lambda MicroVMs, a serverless compute service that provides VM-level isolation with near-instant startup and resume capabilities, built on Firecracker virtualization technology. The service became immediately available in US East (N. Virginia, Ohio), US West (Oregon), Europe (Ireland), and Asia Pacific (Tokyo), offering isolated sandboxes for running user-generated or AI-generated code with state preservation up to 8 hours.

🧠 On July 16, 2026, Moonshot AI released Kimi K3, a mixture-of-experts large language model with 2.8 trillion parameters, of which 104 billion are activated per token from a pool of 896 experts with 16 selected per token, and a 1-million-token context window built on Kimi Delta Attention and Attention Residuals. It debuted at No. 3 on the Artificial Analysis AI leaderboard, behind Anthropic's Claude Fable 5 and OpenAI's GPT-5.6 Sol, and outperformed its competitors on Arena.ai's front-end web development benchmark.

🧠 On July 27, 2026, Moonshot AI published the full Kimi K3 weights on Hugging Face as a 1.4 TB download, making it the largest open-weight model released to that date. The weights shipped under a custom, MIT-derived Kimi K3 licence that permits research, internal, and smaller commercial use but requires a separate commercial agreement with Moonshot once Model-as-a-Service revenue passes \$20 million a year.