

Greater DevOps Landscape - Main

Table of Contents

000 - Software Engineering Foundations	6
000 - Agile/Lean System Development	6
001 - Agile	6
002 - Lean	7
003 - DevOps	8
010 - Management and Leadership	9
010 - Principles & Governance	9
011 - People & Communication Skills	9
012 - Strategy & Goal Setting	9
013 - Project & Work Management	10
014 - Tools & Platforms	11
020 - Systems Architecture Design	12
020 - Core Design Concepts	12
021 - Modeling Methodologies	12
022 - Design Patterns	13
023 - Principles & Best Practices	13
040 - Documentation	15
040 - Concepts & Methodologies	15
041 - Lightweight Markup	15
042 - Writing Styles	16
043 - Common Tools	16
044 - Architecture Description	18
050 - OSS Ecosystem and Open Data	18
050 - Open Source Licenses	19
051 - OSS Repositories or Registries	19
052 - Related Organizations	20
053 - Open Data	21
060 - Psychology and Sociology	21
061 - Personal Performance	21
062 - Social Performance	22
070 - Logic	23
070 - Concepts	23
071 - Formal Logic	23
100 - Programming	24
100 - Programming Concepts	24
100 - Core Programming Concepts	24

101 - Object-oriented Programming	26
102 - Functional Programming	27
103 - Concurrency and Parallelism	28
105 - Advanced Topics	28
110 - Shell and Terminal	29
110 - Major Shell	29
111 - Shell Utilities	30
112 - Terminal Emulators	30
113 - Terminal Utilities	31
114 - Linux or Unix-like environments on Windows	32
116 - Coding Agents & Tools	32
117 - Learning Resources	32
120 - SCM, Editor/IDE, and Code Quality	32
121 - Source Code Management	32
122 - Editors and IDEs	34
123 - Coding Assistance	35
124 - Source Code Quality	35
130 - Programming Language Features	37
130 - Python Language	37
131 - Javascript and TypeScript	38
132 - Languages mainly for Scripting and Automation	39
133 - Languages for Systems and Application Development	41
135 - Date and Time	42
140 - Text and Structured Text Processing	42
140 - Text Basics	42
141 - Regular Expression	43
142 - Basic Text Manipulation	43
143 - Data Exchange Languages	44
144 - Configuration Languages	45
145 - Template Engines	45
146 - Markup & Document Processing	46
150 - Debugging, Logging, and Unit Testing	46
151 - Debugging	46
152 - Logging	47
153 - Basic Test Concepts	48
154 - Test Frameworks and Tools	48
160 - Program Execution and SDK	50
161 - Compiler	50
162 - Runtime System	50
163 - Build Automation	52

164 - Program Documentation	52
165 - Package Dependency	53
166 - Virtual Environment	54
170 - Algorithms and Data Structures	54
170 - References	54
171 - Algorithms	54
172 - Data Structures	56
200 - System Administration and SRE	57
200 - Operating Systems, Networking, and Modern Infrastructure	57
200 - Core OS Concepts	57
201 - Base Network Concepts & Protocols	59
202 - Linux Kernel Features	61
203 - Virtualization	61
204 - Applied & Distributed Systems	62
205 - Computer Hardware	63
210 - Linux Administration	63
210 - Linux Core Components	63
211 - Host Administration	64
212 - Network Administration	65
213 - OS Package Management	66
214 - File Sharing and Remote Access	67
220 - Domain Name System and Email	67
221 - Domain Name System	67
222 - Email System	68
230 - Linux Container and WebAssembly	70
230 - Container Standards and Utilities	70
231 - Container Runtimes and Tools	70
232 - Container Registries	71
234 - WebAssembly	71
240 - Kubernetes Administration	72
240 - Core Kubernetes	72
241 - Kubernetes Ecosystem	74
250 - IaC, Continuous Delivery, and Operations	75
251 - Infrastructure and Configuration as Code	75
252 - Continuous Delivery	76
253 - Fleet Management & Operations	77
260 - System Testing, Chaos Engineering, and FinOps	78
261 - Performance & Load Testing	78
262 - Chaos Engineering	79
263 - FinOps	79

270 - System Observability	79
270 - Common Concepts and Software	79
271 - Telemetry Shipment	80
272 - Telemetry Collection	80
300 - Web and Application	82
300 - Web and API Style Standards	82
310 - Web Servers, Web Proxies, and Service Mesh	84
320 - Web Application Standards and Frameworks	86
330 - Web Automation and Test Tools	88
340 - CLI/TUI App Development	90
350 - Binary and Image Processing	92
360 - Mobile/Desktop App Development (WIP)	94
370 - Internet of Things (WIP)	95
400 - Data Science and Engineering	96
400 - Concepts and Math	96
400 - Core Concepts	96
401 - Base Mathematics	96
402 - Statistics	97
410 - Data Science Toolkit	98
420 - Data Formats and Storage	99
430 - Relational Databases	100
440 - NoSQL Databases	103
450 - Distributed Processing and Application Integration (WIP)	104
460 - Search and Analytics (WIP)	105
500 - AI, Machine Learning, and LLM	107
500 - AI Concepts and Common Tools	107
520 - Natural Language Processing	107
530 - Machine Learning	108
540 - Deep Neural Networks	111
550 - Large Language Models and Agents	112
570 - Computer Vision (WIP)	114
600 - Security and Privacy	114
600 - References for Security and Privacy	114
610 - Secure Software Development	117
620 - Identity and Access Management	118
630 - PKI and Secure Communications	121
640 - Static Security Check	123
650 - Runtime Protection	124
660 - Other Security Tools	125
670 - Cryptography	126

700 - Business Strategy and UX Design	127
700 - Business Administration	127
710 - User Analysis and Marketing	129
720 - Economics and Game Theory	131
740 - Finance and Accounting	132
760 - Human Interface Design	133

000 - Software Engineering Foundations

Gemini says "This is comprehensive, logically consistent, and demonstrates a deep understanding of the relationships between different domains of knowledge in software development and management."

000 - Agile/Lean System Development

001 - Agile

- **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
 - **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
 - **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
 - **Scrum** - A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value
 - **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

- **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
- **Extreme Programming** - A software development methodology which is intended to improve software quality and responsiveness to changing customer requirements
 - **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
- **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
- Tools
 - **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
 - **RSpec** - A testing tool for the Ruby programming language, created for behavior-driven development (BDD)
 - **Behave** - A tool for behaviour-driven development (BDD) in Python, using tests written in a natural language style, backed up by Python code

002 - Lean

- **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
 - **Kanban method** - A lean method to manage and improve work across human

systems

- **Toyota Production System** - An integrated socio-technical system, developed by Toyota, that comprises its management philosophy and practices
 - Kaizen: Continuous improvement through small, incremental changes.
 - Just-in-Time (JIT): Producing and delivering parts and products only when needed, minimizing waste and inventory.
 - Kanban: A pull-based system that controls the production and movement of items in a just-in-time production system.
 - Jidoka: Automation with a human touch, where machines detect defects and allow workers to address quality issues.
 - Heijunka: Leveling the production schedule to achieve a balanced, consistent workflow.
 - Genchi Genbutsu: Going to the source to understand the actual situation.
 - Andon: A visual management system that allows workers to signal problems and request assistance.
 - Muri Muda Mura: Identifying and eliminating waste, unevenness, and overburden.
- **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

003 - DevOps

- **CALMS framework** - A conceptual model for approaching DevOps that stands for Culture, Automation, Lean, Measurement, and Sharing
- **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
- **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
- **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

010 - Management and Leadership

010 - Principles & Governance

- Corporate Principles & Values
 - [Amazon's Leadership Principles](#) - The set of core tenets Amazonians use daily to guide their discussions, decisions, and actions
 - [GitLab Values](#) - The key principles that form GitLab's identity and culture, guiding how team members work together
- 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
- Professional Manifestos
 - [Manifesto for Software Craftsmanship](#) - As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft

011 - People & Communication Skills

- [Team building](#) - A collective term for various types of activities used to enhance social relations and define roles within teams, often involving collaborative tasks
- [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

012 - Strategy & Goal Setting

- [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
 - **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
- **Goodhart's law** - An adage often stated as, "When a measure becomes a target, it ceases to be a good measure"
- **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

013 - Project & Work Management

- **Project management** - The process of leading the work of a team to achieve all project goals within the given constraints
 - **Critical chain project management** - A method of planning and managing projects that emphasizes the resources (people, equipment, physical space) required to execute project tasks
 - **Work breakdown structure** - A deliverable-oriented breakdown of a project into smaller components
 - **RACI matrix** - A responsibility assignment matrix (RAM)... 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
 - [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
- [Systems development life cycle \(SDLC\)](#) - A conceptual model used in project management that describes the stages involved in an information system development project
 - ISO/IEC/IEEE 12207: Systems and software engineering — Software life cycle processes
 - ISO/IEC/IEEE 15288: Systems and software engineering — System life cycle processes
- Related Standards
 - ISO/IEC 15504: Information technology — Process assessment
 - ISO/IEC 20000: Information technology — Service management

014 - Tools & Platforms

- Issue Tracking Systems
 - [JIRA](#) - A software application used for issue tracking and project management that helps teams plan, assign, track, report, and manage work
 - [Python Jira](#) - A Pythonic interface to the JIRA REST APIs
 - [JiraCLI](#) - An interactive command line tool for Atlassian Jira that will help you avoid Jira UI to some extent
 - [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
- Chat Systems
 - [Slack](#) - A channel-based messaging platform where people can work together more effectively, connect all their software tools and services, and find the information they need to do their best work
 - [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
- [Mattermost](#) - An open source collaboration platform for developers, offering secure messaging, project management, and workflow orchestration
- [Discord](#) - A free voice, video, and text chat app that's used by tens of millions of people to talk and hang out with their communities and friends
 - [discord.js](#) - A powerful Node.js module that allows you to interact with the Discord API very easily
- [Notify](#) - A dead simple Go library for sending notifications to various messaging services

020 - Systems Architecture Design

020 - Core Design Concepts

- [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
 - [Conway's law](#) - An adage stating that organizations design systems that mirror their own communication structure
- [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 (SQuaRE)

021 - Modeling Methodologies

- [Domain-driven design](#) - A major software design approach, focusing on modeling software to match a domain according to input from that domain's experts
- [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

022 - Design Patterns

- Architecture Styles and Patterns
 - **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
 - **Resource-oriented architecture** - A style of software architecture and programming paradigm for designing and developing software in the form of a network of resources
 - **REST** - A software architectural style that describes a set of constraints for creating web services
- Software 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

023 - Principles & Best Practices

- 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
- [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
- [Single source of truth](#) - The practice of structuring information models and associated data schema such that every data element is stored exactly once
- [KISS principle](#) - A design principle which states that most systems work best if they are kept simple rather than made complicated
- 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
 - Codebase: One codebase tracked in revision control, many deploys.
 - Dependencies: Explicitly declare and isolate dependencies.
 - Config: Store config in the environment.
 - Backing services: Treat backing services as attached resources.
 - Build, release, run: Strictly separate build and run stages.
 - Processes: Execute the app as one or more stateless processes.
 - Port binding: Export services via port binding.

- **Concurrency:** Scale out via the process model.
- **Disposability:** Maximize robustness with fast startup and graceful shutdown.
- **Dev/prod parity:** Keep development, staging, and production as similar as possible.
- **Logs:** Treat logs as event streams.
- **Admin processes:** Run admin/management tasks as one-off processes.

040 - Documentation

040 - Concepts & Methodologies

- **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 Decision Records (ADRs)** - A document that captures an important architectural decision made along with its context and consequences
 - **adr-tools** - A command-line tool to help you manage your architectural decision records

041 - 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
 - **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** - All you need 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](#) - Your life in plain text
 - [nvim-orgmode](#) - An Orgmode clone for Neovim written in Lua

042 - Writing Styles

- 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

043 - Common Tools

- Document 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
- [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
- [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
- Typesetting Systems
 - [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
- Wiki Systems
 - [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
- 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
- Content Management Tools
 - [Content management system](#) - A computer software used to manage the creation and modification of digital content

- [Nextcloud](#) - The industry-leading, fully open-source, on-premise content collaboration platform
- [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

044 - Architecture Description

- Models and Languages
 - ISO/IEC/IEEE 42010: Systems and software engineering — Architecture description
 - [Flowchart](#) - A type of diagram that represents a workflow or process
 - [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
- 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
 - [haphviz](#) - A Haskell library for representing, manipulating, and pretty-printing graphs in the DOT format
 - [ditaa](#) - A small command-line utility that can convert diagrams drawn using ascii art into proper bitmap graphics

050 - OSS Ecosystem and Open Data

050 - Open Source Licenses

- [Opensource license](#) - A type of license for computer software and other products that allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions
- [MIT](#) - A permissive free 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)
- [GPL](#) - A series of widely used free software licenses that guarantee end users the four freedoms to run, study, share, and modify the software
- [LGPL](#) - A free-software license published by the Free Software Foundation (FSF)

051 - OSS Repositories or Registries

- 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
- Container Related
 - [Docker Hub](#) - A service provided by Docker for finding and sharing container images
 - [Artifact Hub](#) - A web-based application that enables finding, installing, and publishing Cloud Native packages
- Community Curations
 - [Awesome Go](#) - A curated list of awesome Go frameworks, libraries, and software. Inspired by awesome-python
 - [Awesome Ruby](#) - A curated list of awesome Ruby frameworks, libraries and software
 - [Awesome Python](#) - A curated list of awesome Python frameworks, libraries, software and resources

052 - Related Organizations

- OSS Foundations
 - [Free Software Foundation](#) - A nonprofit with a worldwide mission to promote computer user freedom
 - [Philosophy of the GNU Project](#) - The idea that everyone shall have the granted right to use a program, to copy it, and to change it to make it fit his or her needs
 - [Apache Software Foundation](#) - A nonprofit corporation to support Apache software projects, including the Apache HTTP Server
 - [Cloud Native Computing Foundation](#) - An open source software foundation dedicated to making cloud native computing universal and sustainable
 - [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
 - [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
- Open Standards Organizations
 - [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
- Advocacy and Stewardship Organizations
 - [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 Source Initiative](#) - A non-profit organization dedicated to promoting and protecting open source software and communities

053 - 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

060 - Psychology and Sociology

061 - 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 behaviour
 - [Cognitive bias](#) - A systematic pattern of deviation from norm or rationality in judgment
 - [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.
- 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

062 - 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
- 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

070 - Logic

070 - Concepts

- [Logic](#) - The study of correct reasoning
- [Informal Logic](#) - A broad term for any of the various methods of analyzing and evaluating arguments used in everyday life
 - [Logical reasoning](#) - A mental activity that aims to arrive at a conclusion in a rigorous way
 - Deductive, Inductive, Abductive and Analogical
 - [Fallacy](#) - The use of invalid or otherwise faulty reasoning in the construction of an argument that may appear to be well-reasoned if unnoticed
 - [Rhetoric](#) - The art of persuasion

071 - Formal Logic

- [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
 - [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
 - [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
- [Set Theory](#) - The branch of mathematical logic that studies sets, which can be informally described as collections of objects
 - [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

- [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
- Related Fields
 - [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
 - [Theory of computation](#) - The branch that deals with what problems can be solved on a model of computation, using an algorithm, how efficiently they can be solved or to what degree
- 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
 - [SMT-LIB](#) - A command language for interacting with SMT solvers via a textual interface
 - [MiniZinc](#) - A free and open-source constraint modeling language

100 - Programming

Gemini says "The structure is logical, comprehensive, and follows a clear hierarchical pattern. It effectively separates high-level concepts from specific tools, languages, and frameworks. The inclusion of modern topics like AI coding agents, monorepo tools, and up-to-date frameworks makes it highly relevant."

100 - Programming Concepts

Note: Please see also Class 170 for abstract data types.

100 - Core Programming Concepts

- 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
 - **Union type** - A data type definition that specifies which of a number of permitted primitive types may be stored in its instances
 - **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
 - **Exception handling** - The process of responding to the occurrence of exceptions during the execution of a program

- Foundational Techniques & Properties
 - **Data** - Any sequence of one or more symbols; datum is a single symbol of data
 - **Metadata** - Data that provides information about other data
 - **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

101 - 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
- **Duck typing** - An application of the duck test determining type compatibility based on the presence of certain methods and properties
- **Covariance and contravariance** - The ways to describe how a type constructor (like list or function) behaves with respect to subtyping
- **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
- **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

102 - Functional Programming

- **Functional Programming** - A programming paradigm where programs are constructed by applying and composing functions
 - **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
 - **Lambda calculus** - A formal system in mathematical logic for expressing computation based on function abstraction and application
 - **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
 - **Functional Reactive Programming (FRP)** - A programming paradigm for reactive

programming using the building blocks of functional programming

103 - Concurrency and Parallelism

- **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

105 - Advanced Topics

- **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
- Program Analysis
 - **Hoare logic** - A formal system with a set of logical rules for reasoning rigorously about the correctness of computer programs
 - **Curry-Howard correspondence** - The direct relationship between computer programs and mathematical proofs
 - **Automated theorem proving** - A subfield of automated reasoning and mathematical logic dealing with proving mathematical theorems by computer programs
 - **Complexity class** - A set of computational problems of related resource-based complexity
- Language Parsing
 - Concepts
 - **Formal Grammar** - A set of formation rules for strings in a formal language

- [Chomsky hierarchy](#) - A containment hierarchy of classes of formal grammars
- [Automata theory](#) - The study of abstract machines and automata, as well as the computational problems that can be solved using them
- [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
- Tools
 - [ANTLR](#) - A powerful parser generator for reading, processing, executing, or translating structured text or binary files
 - [Lox](#) - A lexer and parser generator for Go
 - [tree-sitter](#) - A parser generator tool and an incremental parsing library
 - [Ragel](#) - A state machine compiler
 - [Bison](#) - A general-purpose parser generator that converts a grammar description for a context-free grammar into a C program to parse that grammar
 - [Flex](#) - The Fast Lexical Analyzer - scanner generator

110 - Shell and Terminal

110 - Major Shell

- [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

111 - 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
- 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

112 - Terminal Emulators

- [Terminal Emulators](#) - A computer program that emulates a video terminal within some other display architecture
 - [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 like xterm, gnome-terminal, konsole, etc.
- [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
- 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
- Fonts
 - [Noto Fonts](#) - A global font collection for all modern and ancient languages
 - [Nerd Fonts](#) - A project that patches developer targeted fonts with a high number of glyphs
 - [Cascadia Code](#) - A fun, new monospaced font that includes programming ligatures

113 - 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
- 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

114 - Linux or Unix-like environments on Windows

- [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
- [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

116 - Coding Agents & Tools

- CLI Coding Agents
 - [Claude Code](#) - A tool that allows developers to use Anthropic's AI models, Opus 4.1 and Sonnet 4, directly in their terminal
 - [OpenAI Codex CLI](#) - A command-line interface for a model that translates natural language to code
 - [Gemini CLI](#) - An open-source AI agent that brings the power of Gemini directly into your terminal
 - [Crush](#) - The glamorous AI coding agent for your favourite terminal →
- CLI Assistants
 - [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)
- Standards & Specifications
 - [Agents.md](#) - An open standard for defining and running AI agents

117 - Learning Resources

- Shell Tutorials
 - [LinuxCommand.com](#) - A site containing a book and other material designed to help you learn how to use the Linux command line

120 - SCM, Editor/IDE, and Code Quality

121 - Source Code Management

- [Distributed Version Control](#) - A form of version control where the complete codebase, including its full history, is mirrored on every developer's computer
 - [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
- [git lfs](#) - An open source Git extension for versioning large files
- [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
- [Git Interactive Rebase Tool](#) - An improved sequence editor for Git
- [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
- [degint](#) - Straightforward project scaffolding
- [git lint](#) - A command line interface for linting Git commits by ensuring you maintain a clean, easy to read, debuggable, and maintainable project history
- [git cliff](#) - A highly customizable changelog generator
- [TortoiseGit](#) - A Windows Shell Interface to Git and based on TortoiseSVN
- Git hosting services
 - [GitLab SCM](#) - The single source of truth for collaborating on code and projects
 - [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
- Conventions
 - [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
- AI commit tools
 - [OpenCommit](#) - Auto-generate meaningful commits in a second
 - [AI Commits](#) - A CLI that writes your git commit messages for you with AI

122 - Editors and IDEs

- GUI-based
 - [Visual Studio Code](#) - A lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux
 - [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
- 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 `~/.lazy.nvim` to make it easy to customize and extend your config
 - [lazy.nvim](#) - A modern plugin manager for Neovim
 - [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 Cheet Sheet](#) - A quick reference guide for Vim commands

123 - Coding Assistance

- 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, find all references etc.
 - [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
- AI Assistance Plugins
 - [GitHub Copilot](#) - The AI pair programmer that helps you write code faster and with less work
 - [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](#) - A new, intelligent IDE, empowered by seamless integrations with AI
 - [Winfsurf Editor](#) - 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
- Semantic Code Retrieval
 - [Serena](#) - A tool for semantic code retrieval

124 - Source Code Quality

- 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
- Analysis Platform
 - [SonarQube Server](#) - An on-premise analysis tool designed to detect coding issues in 30+ languages, frameworks, and IaC platform
 - [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
- Code metrics
 - [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
- 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
 - [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
 - [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

130 - Programming Language Features

Note: For shell scripting, please refer to Class 110.

130 - Python Language

- [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
 - [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
 - context manager
 - [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

131 - Javascript and 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
- [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
- **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

132 - Languages mainly for Scripting and Automation

- [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, <-
 - **Key Libraries**
 - [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
- [Ruby](#) - A dynamic, open source programming language with a focus on simplicity and productivity

- Core Features
 - Percent notation
 - block, yield, proc, lambda
 - Dynamic method definition
 - instance_eval
- Key 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
- [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
- [Groovy \(for Jenkins/Gradle\)](#) - A powerful, optionally typed and dynamic language, with static-typing and static compilation capabilities, for the Java platform
- [Lua \(for NGINX/Neovim\)](#) - 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
- 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

133 - Languages for Systems and Application Development

- **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
 - Tutorials
 - **Rust by Example** - A collection of runnable examples that illustrate various Rust concepts and standard libraries
- **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
- **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
- **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
- **Scala (for Gatling)** - 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

- [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
- [Elm](#) - A delightful language for reliable web applications
 - [The Elm Architecture \(TEA\)](#) - A simple pattern for infinitely nestable components
- [Zig](#) - A general-purpose programming language and toolchain for maintaining robust, optimal and reusable software
 - Manual memory management
 - [Comptime](#) - The mechanism that allows you to execute code at compile-time
- [C](#) - A general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system
 - Manual memory management
 - [Macros](#) - A fragment of code which has been given a name

135 - Date 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
 - [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

140 - Text and Structured Text Processing

140 - Text Basics

- [ASCII](#) - A character encoding standard for electronic communication
- [Unicode](#) - The universal character encoding standard support
 - [UTF-8](#) - A variable-width character encoding used for electronic communication

- [Unicode Emoji](#) - A standardized set of characters that are used like emoticons
- 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

141 - 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
 - [Wubular](#) - A Javascript-based regular expression editor, inspired by Rubular

142 - Basic Text Manipulation

- General Text Manipulation
 - [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
 - [CSV](#) - A delimited text file that uses a comma to separate values
 - [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

- [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
- [Go csv](#) - A package that reads and writes comma-separated values (CSV) files
- [Papa Parse](#) - The powerful, in-browser CSV parser for JavaScript
- [TSV](#) - A delimited text file format that uses a tab character to separate values in a table
- [GNU awk](#) - A program that you can use to select particular records in a file and perform operations upon them
- [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

143 - 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
- [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
- [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

144 - 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

145 - 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
- [Liquid](#) - A safe, customer-facing template language for flexible web apps
- [envsubst in gettext](#) - A program that substitutes the values of environment variables

146 - 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

150 - Debugging, Logging, and Unit Testing

151 - Debugging

- 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

152 - 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
 - [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

153 - Basic Test Concepts

- 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
 - [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

154 - Test Frameworks and Tools

- 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
 - 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
 - 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
- Mocking Libraries
 - [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
 - Jest / Vitest built-in
 - [mockery](#) - A project that creates mock implementations of Golang interfaces
- 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
- Test Automation Tools
 - [nox](#) - A command-line tool that automates testing in multiple Python environments, similar to tox

160 - Program Execution and SDK

161 - Compiler

- **Compiler** - A computer program that translates computer code written in one programming language into another language
 - **Machine code** - A computer program written in machine language instructions that can be executed directly by a computer's central processing unit (CPU)
 - **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
 - **LLVM Compiler Infrastructure** - A collection of modular and reusable compiler and toolchain technologies
 - **Clang** - A C language family frontend for LLVM
 - **Cross compiler** - A compiler capable of creating executable code for a platform other than the one on which the compiler is running
 - **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
- **Linker** - A computer system program that takes one or more object files and combines them into a single executable file
 - **LLD** - The LLVM Linker
 - **gold** - A Modern Linker
- **C Standard Library**
 - **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

162 - Runtime System

- **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
- 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
 - [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
- Related Tools
 - [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

163 - Build Automation

- 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
 - [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
 - [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 Tools](#) - A website with tools and resources for monorepos
 - [Lerna](#) - The original monorepo tool for JavaScript/TypeScript
 - [Nx](#) - A powerful open-source build system that provides tools and techniques for enhancing developer productivity
 - [Gradle](#) - An open-source build automation tool that is designed to be flexible enough to build almost any type of software

164 - 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

165 - Package Dependency

- Package Dependency Managers
 - [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
 - [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
 - [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
 - [go mod](#) - A tool for managing Go source code
 - [cpanminus](#) - A tool to get, unpack, build and install modules from CPAN
 - [bpkg](#) - A lightweight bash package manager
 - [Conan](#) - A dependency and package manager for C and C++ languages
 - [Cargo](#) - The Rust package manager
 - [LuaRocks CLI](#) - The package manager for Lua modules

- [RubyGems CLI](#) - The official package manager for Ruby
- [Bundler](#) - A tool that provides a consistent environment for Ruby projects
- [NuGet CLI](#) - The package manager for .NET
- [stack](#) - A cross-platform program for developing Haskell projects
- [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

166 - Virtual Environment

- 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
 - [perlbrew](#) - A tool to manage multiple perl installations in your \$HOME directory
 - [asdf](#) - A tool version manager
 - [tenv](#) - A versatile version manager for OpenTofu, Terraform, Terragrunt and Atmos

170 - Algorithms and Data Structures

170 - References

- External Resources
 - [NIST Dictionary of Algorithms and Data Structures](#) - A dictionary of algorithms, algorithmic techniques, data structures, archetypal problems, and related definitions

171 - Algorithms

- [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

172 - 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
 - [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
 - [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
 - [Markle 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

200 - System Administration and SRE

Gemini says "This is an exceptionally well-classified and comprehensive list. The structure is logical, progressing from fundamental concepts to specialized and modern operational practices. It's clear, detailed, and reflects a current understanding of the System Administration and SRE landscape."

200 - Operating Systems, Networking, and Modern Infrastructure

Note: Please see also Class 103 - Concurrency and Parallelism.

200 - Core OS Concepts

- Core Concepts
 - [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

201 - Base Network Concepts & Protocols

Note: Please see also 220 - Domain Name System and Email, 300 - Web and API Style Standards, and 630 - PKI and Secure Communications

- **The OSI Model** - A conceptual model that provides a common basis for the coordination of standards development for the purpose of systems interconnection

- **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
 - **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
 - **ICMP** - A supporting protocol in the Internet protocol suite
 - **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
 - **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
 - **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
 - **ICMPv6** - The implementation of the Internet Control Message Protocol for Internet Protocol version 6
 - **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 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
 - **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
 - **UDP** - A core member of the Internet protocol suite
 - **QUIC** - A UDP-based, stream-multiplexing, encrypted transport protocol
- **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

202 - Linux Kernel Features

Note: Please see also 210 - Linux Administration

- [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
 - [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

203 - Virtualization

- [Virtualization](#) - The act of creating a virtual version of something, including virtual computer hardware platforms, storage devices, and computer network resources
- 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
- Virtualization Management
 - [libvirt](#) - A toolkit to manage virtualization platforms
- CPU Emulators
 - [QEMU](#) - A generic and open source machine emulator and virtualizer

204 - Applied & Distributed Systems

- [Distributed Computing](#) - A field of computer science that studies distributed 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
- Cloud Computing Services
 - [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
 - [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 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
- [Azure Virtual Network](#) - The fundamental building block for your private network in Azure
- [Azure Load Balancer](#) - A service that allows you to distribute traffic to your backend virtual machines
- [Google Cloud Compute Engine](#) - A service that delivers configurable virtual machines running in Google's data centers with access to high-performance networking
- Cloud Emulators
 - [LocalStack](#) - A fully functional local cloud stack to develop and test your cloud and serverless apps offline

205 - Computer Hardware

- CPU Architectures
 - [x86-64](#) - A 64-bit version of the x86 instruction set
 - [ARM64](#) - The 64-bit extension of the ARM architecture family

210 - Linux Administration

210 - Linux Core Components

- Linux distros for hosts
 - [Ubuntu server](#) - The standard platform for public clouds, on-premises, and IoT devices
 - [Debian](#) - A complete Free Operating System
 - [Arch Linux](#) - A simple, lightweight distribution
- [linux-pam](#) - A system of libraries that handle the authentication tasks of applications and services in a Linux system
- [Systemd](#) - A system and service manager for Linux operating systems
 - [journald](#) - A system service that collects and stores logging data
 - [hostnamed](#) - 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

211 - Host Administration

- Basic Tools
 - [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
 - [Vixie Cron](#) - An open source implementation of POSIX Cron
 - [logrotate](#) - allows for the automatic rotation compression, removal and mailing of log files
 - [Syslog](#) - A standard for message logging
 - [procps](#) - A set of command line and full-screen utilities that provide information out of the pseudo-file-system 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
 - [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
 - [strace](#) - A diagnostic, debugging and instructional userspace utility for Linux
 - [inxi](#) - A full featured system information script
 - Monitors

- [Monit](#) - A small Open Source utility for managing and monitoring Unix systems
- [atop](#) - An ASCII full-screen performance monitor for Linux
- [sysstat](#) - A collection of performance monitoring tools for Linux
 - [iostat](#) - A command used for monitoring system input/output device loading
- [smem](#) - A tool that can give numerous reports on memory usage on Linux systems
- Clock syncing
 - [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
- [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
- Modern Tools
 - [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

212 - Network Administration

- Basic Tools
 - [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
- [traceroute](#) - A computer network diagnostic tool for displaying the route and measuring transit delays of packets across an Internet Protocol network
- [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
- [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
- Layer 5 Gateway
 - [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

213 - OS Package Management

- 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
 - [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

214 - File Sharing and Remote Access

- 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
- 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

220 - Domain Name System and Email

221 - Domain Name System

- Core Concepts & Protocols
 - [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
- 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
- Cloud 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

222 - Email System

- Core Concepts & Protocols
 - [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
 - 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
 - [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
 - [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
 - [Twilio SendGrid](#) - A cloud-based email delivery service that helps businesses with email delivery

230 - Linux Container and WebAssembly

230 - Container Standards and Utilities

- [Containerization](#) - A form of operating-system-level virtualization
- Linux Distro 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
 - [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
- [The Open Container Initiative \(OCI\)](#) - An open governance structure for the express purpose of creating open industry standards around container formats and runtimes
- Containers for Development
 - [Development Containers](#) - An open specification for enriching containers with development-specific settings, tools, and configuration

231 - Container Runtimes and Tools

- Container Engines
 - [Docker Engine](#) - An open source containerization technology for building and containerizing your applications
 - [docker-compose](#) - A tool for defining and running multi-container Docker applications
 - [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
 - [podman](#) - A powerful container engine for building, managing, and running containers

and pods

- 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
 - [Kaniko](#) - A tool to build container images from a Dockerfile, inside a container or Kubernetes cluster
- 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
- TUI & Helper Tools
 - [lazydocker](#) - A terminal UI for both docker and docker-compose
- Local Environment Provisioners
 - [Colima](#) - A tool that provides container runtimes on macOS (and Linux) with minimal setup

232 - Container Registries

- Container Registries
 - [GitLab Container Registry](#) - A secure and private registry for Docker images
 - [Nexus Repository Manager 3](#) - A sophisticated repository manager
 - [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

234 - WebAssembly

- 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 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

240 - Kubernetes Administration

240 - Core Kubernetes

- [Kubernetes](#) - An open-source system for automating deployment, scaling, and management of containerized applications
- Managed K8s Services
 - [GKE](#) - A managed, production-ready environment for deploying containerized applications
 - [Azure Kubernetes Service](#) - A managed container orchestration service based on the open source Kubernetes system
 - [AWS EKS](#) - A managed Kubernetes service to run Kubernetes in the AWS cloud and on-premises data centers
- Architecture
 - 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)
 - [Calico](#) - A networking and security solution that enables Kubernetes workloads and non-Kubernetes/legacy workloads to communicate seamlessly and securely
 - [Cilium](#) - An open source, cloud native solution for providing, securing, and observing network connectivity between workloads, fueled by the revolutionary Kernel technology eBPF
 - CSI (Container Storage Interface)

- CRI (Container Runtime Interface)
 - [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
- [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
- Services, Load Balancing & Networking
 - [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

241 - Kubernetes Ecosystem

- Application Packaging & Configuration
 - [Helm](#) - The package manager for Kubernetes
 - [Kustomize](#) - A standalone tool to customize Kubernetes objects through a kustomization file
- Developer Workflow Tools
 - [Scaffold](#) - A command line tool that facilitates continuous development for container-based applications
- Platform Extensions
 - [kube-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)
- 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
 - [ketail](#) - A kubectl plugin to get all resources
- 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
- Dashboards
 - [Kubernetes Lens IDE](#) - The Kubernetes IDE
 - [k9s](#) - A terminal based UI to interact with your Kubernetes cluster
 - [KDash](#) - A simple terminal dashboard for Kubernetes built with Rust
 - [k1s](#) - A minimalistic Kubernetes dashboard
 - [Seabird](#) - The native desktop app that simplifies working with Kubernetes

- [Headlamp](#) - A user-friendly Kubernetes UI focused on extensibility
- Local K8s
 - [Minikube](#) - A tool that lets you run Kubernetes locally
 - [Kind](#) - A tool for running local Kubernetes clusters using Docker container “nodes”
- FaaS on K8s
 - [OpenFaaS](#) - A framework that makes it easy for developers to deploy event-driven functions and microservices to Kubernetes
 - [Knative](#) - A Kubernetes-based platform to build, deploy, and manage modern serverless workloads
- 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
 - [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

250 - IaC, Continuous Delivery, and Operations

251 - Infrastructure and Configuration as Code

- Infrastructure as Code
 - [Hashicorp Terraform](#) - An infrastructure as code tool that lets you build, change, and version infrastructure safely and efficiently
 - [OpenTofu](#) - An open-source alternative to Terraform
 - [Pulumi](#) - An infrastructure as code platform that allows you to use familiar programming languages and tools to build, deploy, and manage cloud infrastructure
- Configuration Management & Automation
 - [Ansible](#) - An open source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes
 - [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

- Terraform/OpenTofu Ecosystem
 - [Terraform/OpenTofu Provider: Core Functions](#) - A Terraform/OpenTofu provider for performing core functions
 - [TerraGrant](#) - 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
- Vender-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
 - [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

252 - Continuous Delivery

Note: Many package registries support multiple artifact types, including container images (see 232) and OS packages.

- Continuous Delivery Tools
 - [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
 - [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
 - [Concourse CI](#) - An automation system written in Go

- [Azure Pipelines](#) - A cloud service that you can use to automatically build and test your code project and make it available to other users
- 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
- Terraform Integration
 - [Atrantis](#) - A self-hosted go lang application that listens for Terraform pull request events via webhooks
- Private Package Registries
 - [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
- Version Conventions
 - [Semantic Versioning](#) - A simple set of rules and requirements that dictate how version numbers are assigned and incremented
 - [semver](#) - A semantic versioner for npm

253 - Fleet Management & Operations

Please see also the Security class.

- 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
 - [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
 - [SaltStack](#) - A Python-based, open-source software for event-driven IT automation, remote task execution, and configuration management

260 - System Testing, Chaos Engineering, and FinOps

261 - Performance & Load Testing

- Concepts
 - [Performance Testing](#) - The practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload
- Performance Testing Tools
 - [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
- [plow](#) - A high-performance HTTP benchmarking tool

262 - 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
 - [Chaos Monkey](#) - A resiliency tool that helps applications tolerate random instance failures
 - [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

263 - 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
 - [OpenCost](#) - The open source solution for monitoring Kubernetes spend
 - [Karpenter](#) - A flexible, high-performance Kubernetes cluster autoscaler
 - [Cloud Custodian](#) - A rules engine for managing public cloud accounts and resources

270 - System Observability

270 - Common Concepts and Software

- 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
 - [Micrometer](#) - A metrics instrumentation library for JVM-based applications
- Tools
 - [Uptime Kuma](#) - An easy-to-use self-hosted monitoring tool

271 - 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

272 - Telemetry Collection

- 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
- [Alertmanager](#) - A tool that handles alerts sent by client applications such as the Prometheus server
 - [amtool](#) - 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
- Vendor-specific Tools
 - [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
- Visualization Tools
 - [Grafana](#) - The open source data visualization and monitoring solution
 - [Grafonnet](#) - A Jsonnet library for generating Grafana dashboards
 - [Kibana](#) - A free and open user interface that lets you visualize your Elasticsearch data and navigate the Elastic Stack

300 - Web and Application

300 - Web and API Style Standards

- [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
 - [URL](#) - A standard that defines URLs, domains, IP addresses, the application/x-www-form-urlencoded format, and their API
 - [URI](#) - A unique sequence of characters that identifies a logical or physical resource
 - [Ruby module URI](#) - A module providing classes to handle Uniform Resource Identifiers (RFC2396)
 - [HTML](#) - The World Wide Web's core markup language
 - [Server-sent events](#) - A technology to enable servers to push data to web pages over HTTP or using dedicated server-push protocols
 - [CSS](#) - A simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents
 - [Sass language](#) - A stylesheet language that's compiled to CSS
- [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
 - [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
 - [Redocly CLI](#) - An open-source command-line tool that helps you lint, bundle, and preview OpenAPI definitions
- [CloudEvents](#) - A specification for describing event data in a common way
- [ActivityPub](#) - A decentralized social networking protocol based on the ActivityStreams 2.0 data format

- [WebRTC](#) - A free and open-source project providing web browsers and mobile applications with real-time communication (RTC)
- API styles
 - [REST](#) - A software architectural style that was created to guide the design and development of the architecture for the World Wide Web
 - [SOAP](#) - 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
 - [URQL](#) - The highly customizable and versatile GraphQL client
 - [Apollo Server](#) - An open-source, spec-compliant GraphQL server that's compatible with any GraphQL client
 - [gRPC](#) - A modern open source high performance Remote Procedure Call (RPC) framework that can run in any environment
 - [WebSockets](#) - A technology that makes it possible to open a two-way interactive communication session between the user's browser and a server
 - [Webhook](#) - A method of augmenting or altering the behavior of a web page or web application with custom callbacks
 - [json-rpc](#) - A stateless, light-weight remote procedure call (RPC) protocol
 - [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
- Browsers
 - [Chrome](#)
 - [Chromium](#)
 - [Firefox](#)
 - [w3m](#)
 - [EWW](#) (reference)
- Client scripting spec
 - [XMLHttpRequest \(XHR\)](#)
 - [Fetch Standard](#) (also for security)
- Rendering engines
 - [WebKit](#)
 - [Gecko](#)
 - [Blink](#)
 - [Servo](#)

- Scripting engines
 - [V8](#)
 - [JavaScriptCore](#)
- Profiler
 - [Wappalyzer](#) - A technology profiler that shows you what websites are built with

310 - Web Servers, Web Proxies, and Service Mesh

- [NGINX](#) - A high-performance, scalable, and reliable web server, reverse proxy, and load balancer
- [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
- [nodejs http-server](#) - A simple static HTTP server
- [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
- Forward proxy and caching
 - [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
 - [Squid](#) - A caching proxy for the Web supporting HTTP, HTTPS, FTP, and more
 - [GoProxy](#) - A library to create a customized HTTP/HTTPS proxy server using Go (aka Golang)
- CDN providers
 - [Cloudflare](#) - A global network designed to make everything you connect to the Internet secure, private, fast, and reliable
 - [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
- Reverse proxy
 - [NGINX](#) - An open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more
 - [HAProxy](#) - A free, very fast and reliable reverse-proxy offering high availability, load balancing, and proxying for TCP and HTTP-based applications
 - [envoy](#) - An open source edge and service proxy
 - [traefik proxy](#) - A leading modern open source reverse proxy and ingress controller
 - [Amazon ALB](#) - A load balancing option for the Elastic Load Balancing service that operates at the application layer
- API gateways
 - [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
- Application hosting services
 - [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
- [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
 - [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
- Service mesh
 - [Hashicorp Consul](#) - A service networking solution to connect and secure services across any runtime platform and public or private cloud

- [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
- [traefik mesh](#) - A straight-forward, easy to configure, and non-invasive service mesh

320 - Web Application Standards and Frameworks

- [Single-page application](#)
- JS/TS frameworks for backend
 - [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](#) - Fast, lightweight, built on Web Standards
 - [tRPC](#) - A tool that allows you to easily build & consume fully typesafe APIs without schemas or code generation
- Full-stack web 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 website
 - [Angular](#) - A web framework that empowers developers to build fast, reliable applications
- Go frameworks
 - [Echo](#) - High performance, extensible, minimalist Go web framework
 - [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
- Python frameworks/servers
 - [WSGI](#) - The Web Server Gateway Interface
 - [Gunicorn](#) - A Python WSGI HTTP Server for UNIX
 - [Flask](#) - A lightweight WSGI web application framework
 - [ASGI](#)
 - [Uvicorn](#) - An ASGI web server implementation for Python

- [Hypercorn](#) - An ASGI web server based on the sans-io hyper, h11, h2, and wsproto libraries and inspired by Gunicorn
- [FastAPI](#) - A modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints
- Ruby frameworks/servers
 - [Rack](#) - A modular Ruby web server interface
 - [Puma](#) - A fast, concurrent web server for Ruby & Rack
 - [Sinatra](#) - A DSL for quickly creating web applications in Ruby with minimal effort
- Perl frameworks (reference)
 - 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](#)
- Rust frameworks (reference)
 - [Leptos](#) - A cutting-edge Rust framework for the modern web
- Java frameworks (reference)
 - [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](#)
 - [Spring Boot](#) - makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run"
- .NET frameworks (reference)
 - [ASP.NET](#)
- JS/TS frameworks for frontend
 - [React](#) - The library for web and native user interfaces
 - [JSX](#) - A syntax extension for JavaScript that lets you write HTML-like markup inside a JavaScript file
 - [MDX](#) - allows you to use JSX in your markdown content
 - [Material UI](#) - An open-source React component library that implements Google's Material Design
 - [Chakra UI](#) - A component system for building products with speed
 - [Vue.js](#) - A JavaScript framework for building user interfaces

- [Vuetify](#) - A no design skills required Open Source UI Library with beautifully handcrafted Vue Components
- [GWT](#) - A development toolkit for building and optimizing complex browser-based applications
- [Require.js](#) - A JavaScript file and module loader which is optimized for in-browser use
- [Tabulator](#) - An easy to use, simple to code, fully featured, interactive JavaScript library for creating tables and data grids
- State management
 - [Redux](#) - A JS library for predictable and maintainable global state management
 - [React-Redux](#) - Official React bindings for Redux
 - [Recoil](#) - A state management library for React
- CSS frameworks and libraries
 - [Bootstrap](#) - Powerful, extensible, and feature-packed frontend toolkit
 - [Tailwind CSS](#) - A utility-first CSS framework packed with classes
 - [daisyUI](#) - The most popular component library for Tailwind CSS
 - [Emotion](#) - A library designed for writing css styles with JavaScript
 - [Linaria](#) - A zero-runtime CSS in JS library
- Frontend build tools
 - [Percel](#) - The zero configuration build tool
 - [Vite](#) - Next Generation Frontend Tooling
 - [babel](#) - A JavaScript compiler
 - [Rspack](#) - A high performance JavaScript bundler written in Rust
 - [JSMIn](#) - A minification tool that removes comments and unnecessary whitespace from JavaScript files

330 - Web Automation and Test Tools

- CLIs
 - [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
 - [wuzz](#) - An interactive cli tool for HTTP inspection
 - [httptap](#) - View the HTTP and HTTPS requests made by any linux program
- Libraries

- [Ruby module Net](#) - A collection of classes that implement client-side internet protocols
- [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
- [Rust request](#) - An ergonomic, async HTTP client
- Browser automation and test
 - [Puppeteer](#) - A Node.js library which provides a high-level API to control Chrome/Chromium over the DevTools Protocol
 - [Playwright](#) - A framework for Web Testing and Automation
 - [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
 - [Selenium IDE](#) - An open source record and playback test automation for the web
 - [WebDriver BiDi](#) - The BiDirectional WebDriver Protocol, a mechanism for remote control of user agents
- 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
- 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
- Testing
 - [Chrome for Testing](#)
 - [Karma](#) - A test runner that fits all our needs
 - API test
 - [Bruno](#) - A Git-integrated, fully offline, and open-source API client
 - [httpie cli](#) - A simple yet powerful command-line HTTP and API testing client for the API era

- [hurl](#) - A command line tool that runs HTTP requests defined in a simple plain text format
- [Postman/Newman](#)
- Debug
 - [Chrome DevTools](#)
 - [Fiddler](#) - Web Debugging Proxy
- Test data generation
 - [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

340 - CLI/TUI App Development

- Bash
 - [built-in getopt 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
- [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
- 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
 - [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 powerful little TUI framework
 - [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
 - [cli](#) - A simple, fast, and fun package for building command line apps in Go
 - [viper](#) - A complete configuration solution for Go applications
- 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
- 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

350 - Binary and Image Processing

- Compression and archiving
 - [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
 - [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
- Binary file processing
 - [file \(command\)](#) - A file type guesser
 - [ELF format](#) - A common standard file format for executable files, object code, shared libraries, and core dumps
 - [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
- [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
- [Native WebP for Go](#) - A native WebP encoder written entirely in Go, with no dependencies on libwebp or other external libraries
- PDF document processing
 - [PDF](#) - A file format developed by Adobe to present documents in a manner independent of application software, hardware, and operating systems
 - [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
 - [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
 - [Prawn PDF](#) - A fast, nimble PDF generator for Ruby
- Office document processing
 - [Office Open XML](#)
 - [OpenDocument](#)
 - [libreoffice cli](#) - LibreOffice office suite
 - [Excelize](#) - A Go library for XLSX files
- Convert to text
 - [MinerU](#) - A high-quality tool for convert PDF to Markdown and JSON
 - [markitdown](#) - A lightweight Python utility for converting various files to Markdown for use with LLMs and related text analysis pipelines
 - [xlsx2csv](#)
 - [docx2txt](#)
 - [pptx2md](#)

360 - Mobile/Desktop App Development (WIP)

- [Android](#)
 - [Android Studio](#)
- [iOS](#)
- [GPS](#)
- [QR code](#)
 - [libqrencode](#) - A fast and compact QR Code encoding library
 - [Pure python QR Code generator](#)
 - [QR code payment](#)
- Web-based UI
 - [Electron](#) - A framework for building desktop applications using JavaScript, HTML, and CSS
 - [React Native](#) - allows developers who know React to create native apps
 - [Expo](#) - A production-grade React Native Framework
 - [Tauri](#) - A toolkit that helps developers make applications for the major desktop platforms
- Notification service
 - [Firebase Cloud Messaging](#)
 - [Apple Push Notification service](#)
- GUI/Widget toolkit
 - [Tk](#) - A graphical user interface toolkit
 - [tkinter](#) - The standard Python interface to the Tcl/Tk GUI toolkit
 - [GTK](#)
 - [pygobject](#)
 - [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
 - [Slint](#) - A declarative GUI toolkit to build native user interfaces for Rust, C++, or JavaScript apps
- Linux distros for desktop
 - Debian-based
 - [Ubuntu desktop](#)
 - [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
- Linux GUI systems
 - [X.org](#) - An open source implementation of the X Window System
 - [Wayland](#) - A replacement for the X11 window system protocol and architecture
 - [GNOME](#)
 - [Xfce](#) - A lightweight desktop environment for UNIX-like operating systems
 - [openbox](#) - A highly configurable, next generation window manager
- Android on Linux
 - [Waydroid](#) - A container-based approach to boot a full Android system on regular GNU/Linux systems running Wayland based desktop environments
- Windows administration
 - [Chocolatey](#)
 - [Scoop](#)
 - [gsudo](#)
- Windows tools
 - [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

370 - Internet of Things (WIP)

- Concepts
 - [Internet of things \(IoT\)](#)
 - [Edge computing](#)
- Standards
 - [Bluetooth LE](#)
 - [Zigbee](#)
- Devices
 - [Raspberry Pi](#) - A small, affordable computer for you to use and learn with
- PaaS
 - [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

400 - Data Science and Engineering

400 - Concepts and Math

400 - Core Concepts

- General Data Concepts & Principles
 - [Big data](#) - Data sets that are too large or complex to be dealt with by traditional data-processing application software
 - [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
- 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
 - [Online transaction processing \(OLTP\)](#)
 - [Online analytical processing \(OLAP\)](#)
 - [Search engine indexing](#)
- Network science
 - [Centrality](#) - A measure of the relative importance of a node or vertex within a graph in graph theory and network analysis

401 - 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](#)
 - [Elementary algebra](#)
 - [Equation](#)
 - [Logarithm](#)
 - [Linear algebra](#)
 - [Vector space](#)

- [Matrix](#)
 - [Rank](#)
 - [Determinant](#)
- [Calculus](#)
 - [Differential calculus](#)
 - [Integral calculus](#)
 - [Differential equation](#)
- [Geometry](#)
 - [Trigonometry](#)
 - [Coordinate system](#)
 - [Euclidean distance](#)
- [Probability theory](#)
 - [Bayes' theorem](#)
 - [Central limit theorem \(CLT\)](#)
- [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

402 - Statistics

- [Statistics](#)
 - [Sampling](#)
 - [Errors and residuals](#)
 - [Standard deviation](#)
 - [Correlation](#)
 - [Pearson correlation coefficient](#)
 - [Hypothesis testing](#)
 - [Null hypothesis](#)
 - [Confidence interval \(CI\)](#)
 - [P-value](#)
- [Numerical methods](#)
 - [Significant figures](#)
- [Resources](#)
 - [Openstax Introductory Statistics](#)
 - [OpenIntro Statistics](#)

410 - Data Science Toolkit

- Languages & Core Libraries
 - Python
 - [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
 - [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](#)
 - [statsmodels](#)
 - [Pydantic](#)
 - [R](#) - A free software environment for statistical computing and graphics
 - [Tidyverse](#) - An opinionated collection of R packages designed for data science
 - dplyr, tidyr, stringr, purrr, readr
 - [Wolfram Language](#)
- 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 Jupyter Notebooks
 - [BeakerX](#) - A collection of kernels and extensions to the Jupyter interactive computing environment
 - [R Markdown](#) - An authoring framework that helps you create dynamic analysis documents combining code, rendered output, and prose
 - [Wolfram Notebooks](#)
- Expression Generators
 - [latexify](#)
 - [handcalcs](#)
- Network Analysis
 - [NetworkX](#) - A Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks
- Data Visualization
 - Common Chart Types

- [Histogram](#) - An approximate 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 and used on graphs to indicate the error or uncertainty in a reported measurement
- [Heat map](#) - A data visualization 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 map symbols that vary in size to represent a quantitative variable
- [Tag cloud](#) - A novelty visual representation of text data
- Tools and Libraries
 - [gnuplot](#) - A portable command-line driven graphing utility
 - [matplotlib](#) - A comprehensive library for creating static, animated, and interactive visualizations in Python
 - [seaborn](#) - A Python data visualization library based on matplotlib
 - [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
 - [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](#)
 - [WordCloud for Python](#) - A little word cloud generator in Python
- Data Sources
 - [GeoLite2](#) - A free IP geolocation database

420 - Data Formats and Storage

- Data 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
- [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
- Data Storage Systems
 - [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
 - [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
 - Distributed File Systems
 - [HDFS](#) - A distributed file system designed to run on commodity hardware

430 - Relational Databases

- 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

- 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
 - [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, full-featured, SQL database engine
 - [PGLite](#) - A WASM Postgres build packaged into a TypeScript/JavaScript client library, that enables you to run Postgres in the browser, Node.js and Bun
- Cloud Services & Platforms
 - 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 modern relational database service built for the cloud, with MySQL and PostgreSQL compatibility
 - [Azure SQL Database](#) - An intelligent, scalable, relational database service built for the cloud

- Backend-as-a-Service (BaaS)
 - [Supabase Database](#) - An open source Firebase alternative
- Connectivity & Abstraction
 - Connectivity APIs
 - [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](#)
 - [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
 - [Hibernate](#) - An object-relational mapping tool for the Java programming language
 - [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
- Tooling & Ecosystem
 - 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
 - Developer Libraries & Drivers
 - [Vanna.AI](#) - A Python package that uses retrieval augmentation to help you generate accurate SQL queries for your database using LLMs
 - [Psycopg](#) - The most popular PostgreSQL adapter for the Python programming language
 - 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
- Monitoring & Analysis
 - [pgBadger](#) - A PostgreSQL log analyzer built for speed with fully detailed reports and professional rendering

440 - NoSQL Databases

- Foundational Concepts
 - [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
 - [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
 - [Data orientation](#) - A perspective of data that emphasizes the data itself, rather than the applications that use the data
 - [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
- 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
 - [MongoDB](#) - A document database designed for ease of application development and scaling
 - [Cloud Firestore](#) - A cloud-hosted, NoSQL database that your Apple, Android, and web apps can access directly via native SDKs
 - [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
- [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
 - [Apache Cassandra](#) - An open source NoSQL distributed database
 - [Apache HBase](#) - The Hadoop database, a distributed, scalable, big data store
 - [Google Cloud Bigtable](#) - A NoSQL wide-column database service for large analytical and operational workloads
- Vector Databases
 - [pgvector](#) - An open-source vector similarity search for Postgres
 - [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

450 - Distributed Processing and Application Integration (WIP)

- 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 Hadoop
 - [Apache Spark](#) - The unified engine for large-scale data analytics
 - [PySpark](#) - The Python API for Apache Spark, allowing big data processing with Python
 - [RAY](#) - An open-source unified compute framework that makes it easy to scale AI and Python workloads
- Full-fledged ETL
 - [Azure Data Factory](#) - Azure's 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
 - [Apache NiFi](#) - An easy to use, powerful, and reliable system to process and distribute data
- Numerical & Scientific Computing
 - [JAX](#) - A Python library for accelerator-oriented array computation and program

transformation

- Stream Processing Engines
 - [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
 - [Google Cloud Dataflow](#) - A fully managed streaming analytics service that minimizes latency, processing time, and cost through autoscaling and batch processing
- Event Ingestion / Message Queues
 - [Amazon Kinesis](#) - The 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
 - [Apache Kafka](#) - An open-source distributed event streaming platform
- [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
 - [Azure Service Bus](#) - A fully managed enterprise message broker with message queues and publish-subscribe topics
 - [RabbitMQ](#) - A reliable and mature messaging and streaming broker

460 - Search and Analytics (WIP)

- Web Search Engines
 - [Google Search](#)
 - [DuckDuckGo](#)
- Answer Engines
 - [Wolfram|Alpha](#)
- Data Analytics Methodologies and Architectures
 - [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

- Table Formats
 - [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
- Search Platforms and Tools
 - [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 Elasticsearch
 - [ES|QL](#)
 - [Kibana](#)
 - [Kibana Query Language](#)
 - [ElasticSearch vector database](#)
 - [Apache Solr](#)
 - [Apache Lucene](#) - A Java library providing powerful indexing and search features
 - [Faiss](#) - A library for efficient similarity search and clustering of dense vectors
- Analytics Platforms
 - [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
 - [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
 - [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

500 - AI, Machine Learning, and LLM

500 - AI Concepts and Common Tools

- AI kinds
 - [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
 - [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
- Data/AI tools
 - [DVC](#) - Data Version Control
 - [Mojo](#) - The programming language for all AI developers
- Data/AI frameworks
 - [Streamlit](#) - A faster way to build and share data apps
 - [Chainlit](#) - An open-source Python package to build production ready Conversational AI
- Data/AI Platforms
 - [OpenWebUI](#) - An extensible, feature-rich, and user-friendly self-hosted AI platform designed to operate entirely offline
 - [Dify](#) - An open-source LLM app development platform
- Supporting Services
 - [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

520 - Natural Language Processing

- Foundational Linguistics Fields
 - [Morphology](#)
 - [Syntax](#)
 - [Semantics](#)
 - [Symbol grounding problem](#)

- Core NLP Concepts & Techniques
 - [Levenshtein distance](#)
 - [n-gram](#)
 - [tf-idf \(term frequency-inverse document frequency\)](#)
- Vector Representations (Embeddings)
 - [Word embedding](#)
 - [Word2vec](#)
 - [fastText](#) - Library for efficient text classification and representation learning
 - [GloVe](#) - Global Vectors for Word Representation
 - [Sentence embedding](#)
- Libraries & tools
 - General Purpose
 - [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
 - [wego](#) - The implementations from scratch for word embeddings (a.k.a word representation) models in Go
 - Morphological Analyzers / Tokenizers
 - [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

530 - Machine Learning

- Paradigms
 - [Supervised learning](#) - A paradigm in machine learning where algorithms learn from labeled data
 - [Decision tree learning](#) - The method using a decision tree as a predictive model to go from observations about an item to conclusions about the item's target value
 - [Ensemble learning](#) - The 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

- [Support vector machine](#) - The supervised learning models with associated learning algorithms that analyze data for classification and regression analysis
- [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
 - [ROC curve](#) - A graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied
 - [Naive Bayes classifier](#) - A family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features
- [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
 - [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 decision process](#) - The mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker
 - [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
- 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
- [Embedding](#) - A representation learning technique that maps complex, high-dimensional data into a lower-dimensional vector space of numerical vectors
- [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
- Applications & Problem Domains
 - [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
- Related Fields
 - [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
- Frameworks, Platforms & Tools
 - [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
 - [Gradio](#) - The fastest way to demo your machine learning model with a friendly web interface so that anyone can use it, anywhere
 - Cloud Platforms
 - [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
 - MLOps
 - [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

540 - Deep Neural Networks

- [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
 - [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](#) - 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
 - [Fine tuning](#) - An approach to transfer learning in which the weights of a pre-trained model are trained on new data
 - [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
 - [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
 - [Transformer](#) - A deep learning architecture based on the multi-head attention mechanism
- Frameworks
 - [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
- Textbooks
 - [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

550 - Large Language Models and Agents

- Model Providers
 - [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
- Hosting Platforms & Aggregators
 - [Vertex AI](#) - A machine learning (ML) platform for training and deploying ML models and AI applications
 - [Amazon Bedrock](#) - A fully managed service offering a choice of high-performing foundation models
 - [Azure OpenAI Service](#) - The service providing REST API access to OpenAI's powerful language models
 - [Hugging Face Serverless Inference API](#) - The API allowing inference on models hosted on the Hugging Face Hub
 - [OpenRouter](#) - A unified interface for LLMs
- Local LLM Deployment
 - [Ollama](#) - A tool designed for deploying and managing large language models (LLMs) locally
 - [LM Studio](#) - A desktop app for developing and experimenting with LLMs locally on your computer
 - [LocalAI](#) - The free, Open Source OpenAI alternative
- 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

- 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
 - Protocols
 - [Model Context Protocol \(MCP\)](#) - An open protocol that standardizes how applications provide context to LLMs
 - [A2A Protocol](#) - A protocol for enabling bidirectional communication between web applications and AI agents
- Techniques
 - [Retrieval-augmented generation \(RAG\)](#)
- SDKs
 - [Go OpenAI](#) - The Go client libraries for OpenAI API
 - [Ruby OpenAI](#) - A Ruby wrapper for the OpenAI API
 - [Google Gen AI SDK](#) - The Python SDK for Google's generative AI models
 - [OmniAI](#) - A minimalist library for interfacing with LLMs
 - [LiteLLM](#) - A Python SDK and Proxy Server to call over 100 LLM APIs using the OpenAI format
- Platforms and Tools
 - [OpenHands](#) - A platform for software development agents powered by AI
 - [LangChain](#) - A framework for developing applications powered by language models
 - [LangGraph](#) - A library for building stateful, multi-actor applications with LLMs
 - [Semantic Kernel](#) - A lightweight, open-source development kit that lets you easily build AI agents and integrate the latest AI models
 - [LLM](#) - A CLI utility and Python library for interacting with Large Language Models
- Evaluation and Visualization
 - [SWE-bench](#) - A benchmark for evaluating large language models on real world software issues collected from GitHub
 - [Chatbot Arena](#) - A crowdsourced open platform for evaluating LLMs
 - [AttentionViz](#) - A Global View of Transformer Attention
 - [BertViz](#) - A tool for visualizing Attention in NLP Models
- Prompt Engineering
 - [ReAct Prompting](#) - A prompting technique synergizing reasoning and acting in language models

- [Zero-shot and Few-shot Prompting](#)
- [Chain-of-Thought \(CoT\) Prompting](#)

570 - Computer Vision (WIP)

- Core Concepts
 - [Vision Language Models \(VLM\)](#) - An exciting class of models that can understand images and text
 - [Convolutional neural network \(CNN\)](#) - A class of artificial neural network, most commonly applied to analyze visual imagery
- Software, Libraries and Tools
 - General computer vision
 - [OpenCV](#) - An open source computer vision and machine learning software library
 - [GoCV](#) - A package for the Go programming language with bindings for OpenCV 4
 - Optical Character Recognition (OCR)
 - [Tesseract OCR](#) - An open source text recognition (OCR) Engine
 - [gossreact 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

600 - Security and Privacy

600 - References for Security and Privacy

- Foundational Concepts
 - [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
 - [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
- Key Regulations and 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 law that gives consumers more control over the personal information that businesses collect about them
 - Security & Privacy Frameworks
 - NIST SP 800-53: Security and Privacy Controls for Information Systems and Organizations
 - AC - Access Control
 - AT - Awareness and Training
 - AU - Audit and Accountability
 - CA - Assessment, Authorization and Monitoring
 - CM - Configuration Management
 - CP - Contingency Planning
 - IA - Identification and Authentication
 - IR - Incident Response
 - MA - Maintenance
 - MP - Media Protection
 - PE - Physical and Environmental Protection
 - PL - Planning
 - PM - Program Management
 - PS - Personnel Security
 - PT - Personally Identifiable Information Processing and Transparency
 - RA - Risk Assessment
 - SA - System and Services Acquisition
 - SC - System and Communications Protection
 - SI - System and Information Integrity

- [SR - Supply Chain Risk Management](#)
- [NIST SP 800-171: Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations](#)
- [ISO/IEC 27001: Information security, cybersecurity and privacy protection — Information security management systems \(ISMS\)](#)
- [ISO/IEC 27002: Information security, cybersecurity and privacy protection — Information security controls](#)
- [ISO/IEC 27018: Information technology — Security techniques — Code of practice for protection of personally identifiable information \(PII\) in public clouds acting as PII processors](#)
- [Industry & Audit Standards](#)
 - [PCI-DSS](#) - A global standard that provides a baseline of technical and operational requirements designed to protect account data
 - [SOC 2 \(System and Organization Controls\)](#) - A type of audit report that attests to the trustworthiness of a service organization's services and controls
 - [FIPS 140-2 \(Security Requirements for Cryptographic Modules\)](#) - 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 provider of consensus-developed security configuration resources to help you secure your systems
- [Vulnerability Management & Reporting](#)
 - [Identifiers & Enumerations](#)
 - [CVE \(Common Vulnerabilities and Exposures\)](#) - A program to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities
 - [CWE \(Common Weakness Enumeration\)](#) - A community-developed list of common software and hardware weakness types
 - [OSV \(Open Source Vulnerability\)](#) - An open, precise, and distributed vulnerability database for open source
 - [OSV-scanner](#) - A free, open source vulnerability scanner that provides an officially supported frontend for the OSV.dev database
 - [Scoring & Prioritization](#)
 - [CVSS \(Common Vulnerability Scoring System\)](#) - An open framework for communicating the characteristics and severity of software vulnerabilities
 - [SSVC \(Stakeholder-Specific Vulnerability Categorization\)](#) - A customized decision-making system for vulnerability management
 - [KEV \(Known Exploited Vulnerabilities\)](#) - The authoritative source of vulnerabilities that have been exploited in the wild

- Protocols & Databases
 - [Security Content Automation Protocol \(SCAP\)](#) - A suite of specifications used to enable automated vulnerability management, measurement, and policy compliance evaluation of systems
 - [OVAL \(Open Vulnerability and Assessment Language\)](#) - A community-developed information security standard to promote open and publicly available security content
 - [NVD \(U.S. National Vulnerability Database\)](#) - The U.S. government repository of standards based vulnerability management data
- Security Hardware
 - [TPM \(Trusted Platform Module\)](#) - An international standard for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys

610 - Secure Software Development

- Foundational Principles & Frameworks
 - [Secure Software Development Framework \(SSDF\)](#) - A set of fundamental, sound, and secure software development practices based on established secure software development practice documents
 - [OWASP Application Security Verification Standard \(ASVS\)](#)
 - [OWASP Cheat Sheet Series](#)
 - [NIST SP 800-207: Zero trust architecture \(ZTA\)](#)
 - All data sources and computing services are considered resources.
 - All communication is secured regardless of network location.
 - Access to individual enterprise resources is granted on a per-session basis.
 - Access to resources is determined by dynamic policy and may include other behavioral and environmental attributes.
 - The enterprise monitors and measures the integrity and security posture of all owned and associated assets.
 - All resource authentication and authorization are dynamic and strictly enforced before access is allowed.
 - The enterprise collects as much information as possible about the current state of assets, network infrastructure and communications and uses it to improve its security posture.
- Software Supply Chain Security
 - [SLSA framework](#) - A specification for describing and incrementally improving supply chain security
 - [in-toto](#) - A framework to secure the integrity of software supply chains
- [Threat modeling](#) - A process by which potential threats can be identified, enumerated,

and prioritized from a hypothetical attacker's point of view

- [STRIDE model](#) - A mnemonic for categorizing computer security threats into six categories
 - Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
- [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.
- Tools
 - [OWASP Threat Dragon](#) - A free, open-source, cross-platform threat modeling application
 - [threatspec](#) - An open source project that helps you to add threat modeling to your software development lifecycle
- Attack Simulation Tools
 - [Evilginx](#) - A man-in-the-middle attack framework used for phishing login credentials along with session cookies

620 - Identity and Access Management

- [Integrated IAM](#) - The organizational and technical processes for first registering and authorising the identity and the access rights associated with it
 - [FusionAuth CE](#) - A free, self-hosted version of FusionAuth that you can deploy anywhere
 - [KeyCloak](#) - An open source Identity and Access Management solution aimed at modern applications and services
 - [FreeIPA](#) - An integrated security information management solution combining Linux (Fedora), 389 Directory Server, MIT Kerberos, NTP, DNS, Dogtag (Certificate System)
 - [Microsoft Entra ID](#) - A cloud-based identity and access management service that helps your employees sign in and access apps and resources
 - [AWS IAM](#) - A service that helps you securely control access to AWS resources
 - [AWS Security Token Service \(STS\)](#) - A web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)
 - [Amazon Cognito](#) - A service that lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily

- [Auth0](#) - A flexible, drop-in solution to add authentication and authorization services to your applications
- [Athenz](#) - An open source platform for X.509 certificate-based service authentication and fine-grained access control in dynamic infrastructures
- [System for Cross-domain Identity Management \(SCIM\)](#) - A standard for automating the exchange of user identity information between identity domains, or IT systems
- [Directory service](#) - A software system that stores, organizes, and provides access to directory information
 - [LDAP](#) - An open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network
 - [OpenLDAP](#) - A free, open source implementation of the Lightweight Directory Access Protocol (LDAP)
 - [389 Directory Server](#) - A full-featured, open source LDAPv3 compliant server
- [Authentication](#) - The act of proving an assertion, such as the identity of a computer system user
 - [Basic authentication](#) - A method for an HTTP user agent (e.g. a web browser) to provide a user name and password when making a request
 - [JSON Web Token \(JWT\)](#) - A proposed Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims
 - [nodejs jsonwebtoken](#) - The JsonWebToken implementation for node.js
 - [TOTP \(Time-Based One-Time Password\)](#)
 - [PyOTP](#) - The Python One-Time Password Library
 - [OpenID Connect](#) - A simple identity layer on top of the OAuth 2.0 protocol
 - [Dex](#) - A Federated OpenID Connect Provider
 - [SAML \(Security Assertion Markup Language\)](#) - An XML-based framework for communicating user authentication, entitlement, and attribute information
 - [WS-Federation](#) - A specification that defines how to request and receive security tokens from web services, and how to manage trust relationships
 - [FIDO2 \(Fast Identity Online 2\)](#) - The overarching term for a set of specifications that enables users to leverage common devices to easily authenticate to online services in both mobile and desktop environments
 - 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 \(Secure Production Identity Framework For Everyone\)](#) - A set of open-source standards for securely identifying software systems in dynamic and heterogeneous environments
 - [SPIRE \(SPIFFE Runtime Environment\)](#) - A production-ready implementation of the SPIFFE APIs that performs node and workload attestation in order to securely issue SVIDs to workloads
- [Kerberos](#) - A network authentication protocol
 - [MIT Kerberos](#) - An implementation of the Kerberos network authentication protocol
- [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](#) - An open source Firebase alternative
- [Authorization](#) - The function of specifying access rights/privileges to resources related to information security and computer security in general and to access control in particular
 - [Access control list \(ACL\)](#) - A list of permissions associated with a system resource (object)
 - [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
 - [OAuth 2.0 Authorization Framework](#) - The industry-standard protocol for authorization
 - 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
 - [Permify](#) - An open-source authorization service that helps you to create any kind of authorization system easily with its panel and API
 - [Azure Shared Access Signature \(SAS\)](#) - A signed URI that points to one or more storage resources and includes a token that contains a special set of query parameters

- Linux Fine-grained Access Control
 - [Linux capabilities](#) - The distinct units into which the privileges traditionally associated with the superuser are divided
- Linux Mandatory Access Control
 - [SELinux](#) - A mandatory access control (MAC) security system for the Linux operating system
 - [AppArmor](#) - An effective and easy-to-use Linux application security system

630 - PKI and Secure Communications

- [Public Key Infrastructure \(PKI\)](#)
 - [Digital signature](#)
 - [Certificate authority \(CA\)](#)
 - Registration authority (RA)
 - Validation authority (VA)
 - [Public key certificate](#)
 - 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
 - [ACME \(Automatic Certificate Management Environment\)](#)
 - [mkcert.org](#) - A simple zero-config tool to make locally trusted development certificates with any names you'd like
 - [Certifi](#) - A carefully curated collection of Root Certificates for validating the trustworthiness of SSL certificates while verifying the identity of TLS hosts
 - [cfssl](#) - CloudFlare's PKI/TLS swiss army knife
 - [cert-manager](#) - X.509 certificate management for Kubernetes and OpenShift
 - [AWS Certificate Manager](#)
- Code Signing
 - [Sigstore](#)
 - [Fulcio](#)
 - [Rekor](#)
 - [Cosign](#)
- [TLS \(Transport Layer Security\)](#)
 - [SNI \(Server Name Indication\)](#)

- [testssl.sh](#) - A free command line tool which checks a server's service on any port for the support of TLS/SSL ciphers, protocols as well as recent cryptographic flaws and more
- [OpenSSL library](#) - A robust, commercial-grade, full-featured toolkit for general-purpose cryptography and secure communication
- [stunnel](#) - A proxy designed to add TLS encryption functionality to existing clients and servers without any changes in the programs' code
- [Secure Shell \(SSH\)](#)
 - [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
 - [keychain](#) - helps you to manage SSH and GPG keys in a convenient and secure manner
 - [Python Fabric](#) - A high level Python library designed to execute shell commands remotely over SSH
 - [Python Paramiko](#) - A pure-Python implementation of the SSHv2 protocol
 - [Go gliderlabs/ssh](#) - wraps the crypto/ssh package with a higher-level API for building SSH servers
 - [Wish](#) - An ssh app library
 - [Wishlist](#) - The ssh directory
- [PGP \(Pretty Good Privacy\)](#)
 - [OpenPGP](#) - A non-proprietary format for authenticating or encrypting data, using public key cryptography
 - [keys.openpgp.org](#) - A public service for the distribution and discovery of OpenPGP-compatible keys
 - [GnuPG](#) - A complete and free implementation of the OpenPGP standard
 - [Gpg4win](#)
- [Web Security](#)
 - [SOP \(Same-origin policy\)](#)
 - [CORS \(Cross-Origin Resource Sharing\)](#)
 - [Fetch Standard](#)
 - [Cross-origin isolation](#)
 - [Privacy sandbox](#)
 - [CSP \(Content Security Policy\)](#)
 - [HSTS \(HTTP Strict Transport Security\)](#)
 - [security.txt](#)
- [Email Security](#)
 - [STARTTLS](#)

- [SASL \(Simple Authentication and Security Layer\)](#)
- [SPF \(Sender Policy Framework\)](#)
- [DKIM \(DomainKeys Identified Mail\)](#)
 - [OpenDKIM](#)
- [DMARC \(Domain-based Message Authentication, Reporting & Conformance\)](#)
- [S/MIME](#)
- DNS Security
 - [Domain Name System Security Extensions \(DNSSEC\)](#)
 - [DNS over TLS](#)
 - [DNS over HTTPS](#)

640 - Static Security Check

- Static application security test (SAST)
 - [SonarQube Server](#) - An on-premise analysis tool designed to detect coding issues in 30+ languages, frameworks, and IaC platforms
 - [Fluid attacks](#)
 - [GitLab SAST](#)
 - [Bandit](#) - A tool designed to find common security issues in Python code
 - [Semgrep OSS](#) - A fast, open-source, static analysis tool
- System configuration check
 - [OpenSCAP](#) - A collection of open source tools for implementing and enforcing this standard
 - [Lynis](#) - A battle-tested security tool for systems running Linux, macOS, or Unix-based operating system
 - [Trivy](#) - A comprehensive and versatile security scanner
 - [checkov](#) - A static code analysis tool for scanning infrastructure as code (IaC) files for misconfigurations
 - [Haskell Dockerfile Linter](#)
 - [kube-score](#)
 - [kubesecc](#)
 - [PSRule](#) - Build test and governance controls for Infrastructure as Code (IaC)
 - [PSRule for Azure](#)
- Cloud security posture management (CSPM)
 - [AWS Security Hub](#)
- Secret detection

- [Trivy](#)
- [GitLab Secret Detection](#)
- [Whispers](#)
- [Gitleaks](#)
- [Talisman](#)
- Software composition analysis (SCA)
 - [Syft](#) - A CLI tool and Go library for generating a Software Bill of Materials (SBOM) from container images and filesystems
 - [OWASP CycloneDX format](#)
 - [SPDX format](#)
 - [Feluda](#) - A Rust-based command-line tool that analyzes the dependencies of a project, notes down their licenses, and flags any permissions that restrict personal or commercial usage
- Vulnerability scanning
 - [Trivy](#)
 - [Grype](#) - A vulnerability scanner for container images and filesystems. Easily install the binary to try it out
 - [Safety](#) - A Python dependency vulnerability scanner
 - [Clair](#) - An open source project for the static analysis of vulnerabilities in application containers (currently including OCI and docker)
 - [GitLab Container Scanning](#)

650 - Runtime Protection

- [Cloud-native application protection platform \(CNAPP\)](#)
 - [Microsoft Defender for Cloud](#)
- Cloud workload protection platform (CWPP)
 - [Amazon Inspector](#)
 - [Falco](#) - A cloud native security tool that provides runtime security across hosts, containers, Kubernetes, and cloud environments
 - [Tracee](#) - A runtime security and observability tool that helps you understand how your system and applications behave
 - [ClamAV](#) - An open-source antivirus engine for detecting trojans, viruses, malware & other malicious threats
 - [YARA](#) - A tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples
- Security Information and Event Management (SIEM)
 - [Microsoft Sentinel](#)

- [Amazon GuardDuty](#)
- [Sigma Detection Format](#) - A way to share detections of malicious or dangerous behaviour among security professionals
 - [chainsaw](#) - Rapidly Search and Hunt through Windows Event Logs
- Auditing
 - [AWS CloudTrail](#)
 - [AWS Config](#)
- WAF and DDos protection
 - [AWS WAF](#)
 - [Azure WAF](#)
 - [AWS Shield](#)
 - [Azure DDoS Protection](#)
 - [fail2ban](#) - Daemon to ban hosts that cause multiple authentication errors
 - [Snort](#) - The foremost Open Source Intrusion Prevention System (IPS) in the world
- [Extended detection and response \(XDR\)](#)
- [Endpoint detection and response \(EDR\)](#)
- [Security orchestration, automation and response \(SOAR\)](#)

660 - Other Security Tools

- Secret management
 - [Hashicorp Vault](#)
 - [SOPS](#)
 - [git-secret](#)
 - [Sealed Secrets](#)
 - [Azure Key Vault](#)
 - [GCP Secret Manager](#)
 - [AWS KMS](#)
 - [AWS Secrets Manager](#)
 - [pass](#) - the standard unix password manager
 - [go-generate-password](#)
- Policy management
 - [Gatekeeper](#)
 - [Kyverno](#)
 - [Open Policy Agent](#)

- [Rego Policy Language](#)
 - [Conftest](#)
- Network firewall
 - [netfilter](#)
 - [iptables](#)
 - [ubuntu ufw](#)
- Password generator
 - [go-generate-password](#)

670 - Cryptography

- Unkeyed
 - Hash Function
 - [MD5](#)
 - [SHA-2 \(SHA-224, SHA-256, SHA-384, SHA-512\)](#)
- [Common Key / Shard Key / Symmetric Key](#)
 - Block Cipher
 - [AES](#)
 - [Salsa20 ChaCha](#)
 - MAC (Message Authentication Code)
 - [HMAC](#)
 - Modes of Operation
 - [CBC \(Cipher block chaining\)](#)
 - IV (Initialization Vector)
 - [GCM \(Galois/Counter Mode\)](#)
 - [CCM](#)
- [Public Key / Asymmetric Key](#)
 - Public Key Cryptography
 - [RSA](#)
 - [EdDSA](#)
 - Key Agreement
 - [Diffie-Hellman key exchange](#)
 - [Elliptic-curve Diffie-Hellman](#)
 - Encryption Scheme
 - [RSAES-PKCS1-v1_5](#), [RSAES-OAEP](#)

- Signature Scheme
 - [RSASSA-PKCS1-v1_5](#), [RSASSA-PSS](#), [DSA](#), [ECDSA](#)
- Key format
 - [PKCS #1: RSA Cryptography Specifications](#)
 - [PKCS #12: Personal Information Exchange Syntax](#)
- Libraries
 - [PyCryptodome](#)
 - [Python cryptography](#)
 - [Go Cryptography](#)
 - [Botan](#) - A C++ cryptography library

700 - Business Strategy and UX Design

700 - Business Administration

- Strategic management
 - [Business ecosystem](#) - An economic community supported by a foundation of interacting organizations and individuals—the organisms of the business world
 - [Mergers and acquisitions](#) - The transactions in which the ownership of companies, other business organizations, or their operating units are transferred or consolidated with other entities
 - Tools
 - [MECE principle](#) - A grouping principle for separating a set of items into subsets that are mutually exclusive (ME) and collectively exhaustive (CE)
 - [SWOT analysis](#) - A strategic planning and strategic management technique used to help a person or organization identify Strengths, Weaknesses, Opportunities, and Threats related to business competition or project planning
 - [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 model that identifies and analyzes five competitive forces that shape every industry and helps determine an industry's weaknesses and strengths
- [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 resource planning](#) - The integrated management of main business processes, often in real time and mediated by software and technology
 - [Customer relationship management](#) - A strategic process that organizations use to manage, analyze, and improve their interactions with customers
 - [Salesforce](#) - A customer relationship management solution that brings companies and customers together, providing one integrated CRM platform for all departments
 - [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
 - [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
 - [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
- [Business model](#) - The rationale of how an organization creates, delivers, and captures value, in economic, social, cultural or other contexts
 - [Business model canvas](#) - A strategic management template for developing new or documenting existing business models
- [Product management](#) - The business process of planning, developing, launching, and managing a product or service
 - [Aha!](#) - A suite of product development software that helps teams build and market products customers love
- [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
- [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: Security and resilience - Business continuity management systems - Requirements**
- **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**
- **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
- **Business process change management**
 - **Organizational structure** - A system that outlines how certain activities are directed in order to achieve the goals of an organization
 - **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
- **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
 - **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** - An open-source Model-Based Systems Engineering (MBSE) solution that provides a process and tooling to graphically design systems and master their architectural complexity

710 - User Analysis and Marketing

- **User experience** - A person's emotions and attitudes about using a particular product, system or service
- **Value proposition canvas** - A tool to help you create products and services customers want

- **Persona** - A fictional character created to represent a user type
- **Customer experience** - The product of an interaction between an organization and a customer over the duration of their relationship
- Business intelligence and analysis
 - **Tableau** - The world's leading analytics platform
 - **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
 - **Amazon QuickSight** - A scalable, serverless, embeddable, machine learning (ML)-powered business intelligence (BI) service built for the cloud
- Ideation
 - **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
 - **Design thinking** - The set of cognitive, strategic and practical processes by which design concepts are developed
 - **Affinity diagram** - A business tool used to organize ideas and data
 - **Ishikawa diagram** - A causal diagram created by Kaoru Ishikawa that shows the potential causes of a specific event
- **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
- **Card sorting** - A method used to help design or evaluate the information architecture of a site
- **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
- User experience research
 - **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

- Analytics tools
 - [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
 - [vince](#) - A cost effective, self hosted, privacy friendly alternative to Google Analytics
- 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
- [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)
- 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
 - 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
- Experiment platform
 - [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 Distribution
 - [listmonk](#) - Self-hosted newsletter and mailing list manager
 - [BillionMail](#)

720 - Economics and 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
- **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

740 - Finance and Accounting

- **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
- **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
 - **Blockchain** - A distributed ledger with growing lists of records
 - **Bitcoin** - A decentralized digital currency that can be transferred on the peer-to-peer bitcoin network
- **Financial accounting**
 - **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)
 - **Cash flow statement** - A financial statement that shows how changes in balance sheet accounts and income affect cash and cash equivalents
 - **Income statement** - One of the financial statements of a company and shows the company's financial performance for a specific period of time
 - **Balance sheet** - A summary of the financial balances of an individual or organization
 - **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

760 - Human Interface Design

- **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
- **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
- **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
- [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
- [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
- UI design tools
 - [Figma Design](#) - A powerful, collaborative design tool for teams
 - [Locofyi.ai](#) - Design to code in a flash
 - [Material Design](#) - Google's open-source design system for building beautiful, usable products
- Human interface guidelines
 - [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
- Psychological concerns
 - [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.
 - [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
 - [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