

How to improve your vocabulary quickly

✓ 1 > Listen  and read  at the same time

✓ 2 > Listen  without reading  .

✓ 3 > Test your comprehension:
listen the vocabulary alone to remember the meaning
and repeat each word

If necessary, you have the translation at the end.

IT ENGINEERING

FIRST PART

- **Software Engineering:** Software engineering is crucial for developing robust and scalable applications using systematic engineering principles.
- **Programming Language:** Choosing the right programming language is essential to express logic and instructions effectively in software development.
- **Algorithm:** Algorithms are the step-by-step procedures or formulas for solving problems and performing tasks in programming.
- **Code:** Writing clean and efficient code is essential for creating software that is easy to understand and maintain.
- **Debugging:** Debugging is the process of identifying and fixing errors or bugs in the code to ensure proper functionality.
- **Version Control:** Version control systems like Git help track changes in code, enabling collaboration and managing code history.
- **Database:** Databases store and manage data, providing a structured and efficient way to retrieve and manipulate information.
- **Data Structure:** Understanding data structures is essential for organizing and storing data effectively in software applications.

- **API (Application Programming Interface):** An API defines how software components should interact, facilitating communication between different systems.
- **Backend:** The backend of a software application manages server-side operations and data processing.
- **Frontend:** The frontend is responsible for the user interface and how users interact with the application.
- **User Interface (UI):** UI design focuses on creating visually appealing and user-friendly interfaces for applications.
- **User Experience (UX):** Improving UX ensures that users have a positive and satisfying interaction with the software.
- **Framework:** Frameworks provide a structured foundation for development, offering pre-built components and tools to streamline coding.
- **Integration Testing:** Integration testing ensures that different components of a system work seamlessly together.
- **Deployment:** Deployment involves releasing and making a software application available for users.
- **Agile:** Agile methodologies prioritize adaptability, collaboration, and customer feedback in the software development process.
- **Scrum:** Scrum is an agile framework that emphasizes iterative development, collaboration, and adaptability.
- **DevOps:** DevOps practices focus on automating and improving collaboration between development and operations teams.
- **Artificial Intelligence (AI):** AI involves creating systems that can perform tasks requiring human-like intelligence.
- **Machine Learning:** Machine learning enables systems to learn and improve from data without explicit programming.
- **Cybersecurity:** Cybersecurity measures are essential for protecting software and systems from unauthorized access and attacks.
- **Cloud Computing:** Cloud computing provides on-demand access to computing resources over the internet.
- **Big Data:** Big Data involves processing and analyzing large and complex datasets to extract insights.

- **Virtualization:** Virtualization allows multiple virtual instances of operating systems or applications to run on a single physical server.
- **Networking:** Networking involves designing, implementing, and managing computer networks for communication.
- **Debug:** Debugging is the process of identifying and resolving errors or bugs in the code.
- **Patch:** Applying patches is necessary to update and secure software by fixing vulnerabilities or adding new features.
- **Encryption:** Encryption is used to secure sensitive data by converting it into a coded format.
- **Open Source:** Open-source software allows users to view, modify, and distribute the source code freely.
- **Scalability:** Scalability ensures that a software application can handle growth and increased demand.
- **Code Review:** Code reviews involve the systematic examination of code by peers to ensure quality and adherence to best practices.
- **Repository:** A code repository is a centralized location for storing and managing version-controlled source code.
- **Continuous Integration (CI):** CI automates the integration of code changes into a shared repository, ensuring early issue detection.
- **Continuous Deployment (CD):** CD automates the deployment process, delivering new code changes to production efficiently.
- **Full Stack:** Full-stack developers are proficient in both frontend and backend development.
- **Responsive Design:** Responsive design ensures that applications adapt to different devices and screen sizes.
- **Backend as a Service (BaaS):** BaaS provides pre-built backend services, allowing developers to focus on frontend development.
- **Frontend Framework:** Frontend frameworks provide tools and structures for building interactive user interfaces.
- **User Authentication:** User authentication verifies the identity of users accessing a system.
- **Library:** Libraries offer pre-written code and functionalities to speed up development.

- **Dependency Injection:** Dependency injection is a design pattern where dependencies are provided externally to a component.
- **Runtime:** Runtime refers to the period when a program is actively running and executing tasks.
- **Serverless:** Serverless architecture allows developers to build and run applications without managing server infrastructure.
- **Microservices:** Microservices architecture involves developing a system as a collection of small, independent services.
- **Containerization:** Containerization involves encapsulating applications and dependencies into containers for consistent deployment.
- **Docker:** Docker is a platform that simplifies the deployment of applications within containers.
- **Kubernetes:** Kubernetes is an open-source container orchestration platform for managing containerized applications.
- **Scalability:** Scalability is the ability of a system to handle increased workload by adding resources.
- **Load Balancing:** Load balancing distributes network traffic across multiple servers to ensure even resource utilization.
- **Failover:** Failover is the process of switching to a backup system to maintain continuous operation in case of a failure.
- **Concurrency:** Concurrency is the ability of multiple tasks to execute in overlapping time periods.
- **Parallelism:** Parallelism involves simultaneously executing multiple tasks for improved performance.
- **Multithreading:** Multithreading allows multiple threads within a single process to execute concurrently.
- **Multiprocessing:** Multiprocessing involves using multiple processors or cores to execute tasks concurrently.
- **API Endpoint:** An API endpoint is a specific URL where an API can be accessed.
- **RESTful:** RESTful design follows principles for building scalable and stateless networked applications.
- **SOAP (Simple Object Access Protocol):** SOAP is a protocol for exchanging structured information in web services.

- **JSON (JavaScript Object Notation):** JSON is a lightweight data-interchange format commonly used in web development.
- **XML (eXtensible Markup Language):** XML is a markup language for encoding documents in a format that is both human-readable and machine-readable.
- **Regular Expression:** Regular expressions are powerful patterns used for searching, matching, and manipulating text.
- **Dependency Management:** Dependency management involves handling external libraries and components that a software project relies on.
- **CI/CD Pipeline:** The CI/CD pipeline automates the processes of Continuous Integration and Continuous Deployment in software development.
- **Unit Testing:** Unit testing involves testing individual units or components of a software application.
- **Integration Testing:** Integration testing ensures that different components of a system work together as intended.
- **End-to-End Testing:** End-to-End testing evaluates the complete workflow of an application from start to finish.
- **Refactoring:** Refactoring is the process of restructuring existing computer code without changing its external behavior.
- **Code Smell:** Code smell refers to any characteristic in the source code that could indicate a deeper problem.
- **Legacy Code:** Legacy code refers to outdated or obsolete code that may need updates.
- **Technical Debt:** Technical debt represents the cost of delayed improvements or fixes in software development.
- **Codebase:** The codebase is the complete body of source code for a software project.
- **Code Repository:** A code repository is a storage space for version-controlled source code.
- **Git:** Git is a distributed version control system widely used in software development.
- **Mercurial:** Mercurial is another version control system similar to Git.
- **Continuous Monitoring:** Continuous monitoring tracks the performance of systems in real-time.
- **Logging:** Logging records events and activities within an application for analysis.
- **Metrics:** Metrics provide quantitative measures of system performance.

- **Performance Tuning:** Performance tuning involves optimizing a system for better efficiency.
- **Profiling:** Profiling analyzes program behavior to identify performance bottlenecks.
- **Code Review:** Code review ensures code quality through peer evaluation.
- **Pair Programming:** Pair programming involves two developers working together on the same code.
- **Agile Manifesto:** The Agile Manifesto outlines principles for agile software development.
- **Sprint:** A sprint is a time-boxed iteration in the Agile development process.
- **Backlog:** The backlog is a prioritized list of tasks yet to be completed.
- **Burn Down Chart:** A burn-down chart visualizes work completed versus remaining work over time.
- **JIRA:** JIRA is a popular project management and issue tracking tool.
- **Confluence:** Confluence is a collaboration tool used for documentation and project management.
- **Epic:** An epic is a large user story that can be broken down into smaller tasks.
- **User Story:** A user story describes a piece of functionality from an end-user perspective.
- **Wireframe:** A wireframe is a basic visual representation of a web page or application.
- **Prototyping:** Prototyping involves creating a preliminary version of a product for testing.
- **MVP (Minimum Viable Product):** MVP is the smallest version of a product that satisfies customer needs.
- **AB Testing:** AB testing compares two versions of a web page to determine the more effective one.
- **DevSecOps:** DevSecOps integrates security practices into the DevOps pipeline.
- **Cryptography:** Cryptography involves securing communication through encryption techniques.
- **Blockchain:** Blockchain is a distributed ledger technology used for secure and transparent transactions.
- **Smart Contract:** A smart contract is a self-executing contract with the terms of the agreement written in code.

- **Frontend Framework:** A frontend framework provides tools and libraries for building user interfaces.
- **Backend Framework:** A backend framework simplifies server-side development.
- **Static Analysis:** Static analysis examines code without executing it to identify potential issues.
- **Dynamic Analysis:** Dynamic analysis analyzes code during execution to find runtime issues.
- **Load Testing:** Load testing assesses how a system performs under specific conditions, measuring its ability to handle a particular load.
- **Stress Testing:** Stress testing evaluates the system's robustness by subjecting it to extreme conditions or high loads.
- **Capacity Planning:** Capacity planning involves estimating and ensuring that a system has enough resources to handle expected loads.
- **Caching:** Caching improves performance by storing frequently accessed data for faster retrieval.
- **Latency:** Latency is the time delay between a request and its response in a system.
- **WebSockets:** WebSockets enable real-time, bidirectional communication between clients and servers.
- **REST API:** A REST API provides a set of rules for building web services, allowing communication between different systems.
- **GraphQL:** GraphQL is a query language for APIs, providing a more efficient and flexible alternative to REST.
- **OAuth:** OAuth is an authentication protocol that enables secure authorization between applications.
- **JWT (JSON Web Token):** JWT is a compact, URL-safe means of representing claims between two parties.
- **Single Sign-On (SSO):** SSO allows users to access multiple applications with a single set of login credentials.
- **Two-Factor Authentication (2FA):** 2FA adds an extra layer of security by requiring two forms of identification for access.
- **Cross-Origin Resource Sharing (CORS):** CORS is a security feature allowing or restricting resource requests between different origins.

- **OWASP:** OWASP (Open Web Application Security Project) provides resources for improving software security.
- **Zero-Day Vulnerability:** A zero-day vulnerability is a software flaw that is unknown to the vendor and exploited by attackers.
- **Penetration Testing:** Penetration testing assesses system security by simulating cyberattacks to identify vulnerabilities.
- **Incident Response:** Incident response involves managing and mitigating the impact of security incidents.
- **Data Migration:** Data migration is the process of transferring data from one system to another.
- **ETL (Extract, Transform, Load):** ETL processes involve extracting, transforming, and loading data from one system to another.
- **Data Warehousing:** Data warehousing involves collecting, storing, and managing data from different sources for analysis.
- **Business Intelligence (BI):** BI involves using data analysis tools to make informed business decisions.
- **DevOps Tools:** DevOps tools automate and streamline the software development and operations processes.
- **Container Orchestration:** Container orchestration manages the deployment, scaling, and operation of containerized applications.
- **Infrastructure as Code (IaC):** IaC involves managing and provisioning infrastructure using code and automation.
- **Server Farm:** A server farm is a collection of servers working together to provide services.
- **VPN (Virtual Private Network):** A VPN secures and encrypts communication over a public network.
- **Firewall:** A firewall protects a network by controlling and monitoring incoming and outgoing traffic.
- **Intrusion Detection System (IDS):** An IDS monitors network or system activities for potential security threats.
- **Distributed Systems:** Distributed systems involve multiple interconnected computers working together to achieve a common goal.

- **Micro Frontends:** Micro frontends is an architectural pattern for developing frontend applications in a modular and independent way.
- **Chaos Engineering:** Chaos engineering involves intentionally introducing failures to test and improve system resilience.
- **Dark Mode:** Dark mode is a user interface option with a dark color scheme, reducing eye strain in low-light conditions.
- **Chatbot:** A chatbot is a computer program designed to simulate conversation with users.
- **Natural Language Processing (NLP):** NLP enables computers to understand, interpret, and generate human language.
- **Predictive Analytics:** Predictive analytics uses data and statistical algorithms to make predictions about future events.
- **Robotic Process Automation (RPA):** RPA automates repetitive tasks using software robots.
- **Serverless Computing:** Serverless computing allows developers to focus on writing code without managing server infrastructure.
- **Low-Code Development:** Low-code development involves using visual interfaces to create applications with minimal hand-coding.
- **API Gateway:** An API gateway manages and controls access to APIs, providing security and scalability.
- **Server-Side Rendering (SSR):** SSR generates HTML on the server before sending it to the client, improving page load times.
- **Client-Side Rendering (CSR):** CSR renders HTML on the client-side, enhancing interactivity but potentially increasing load times.
- **Immutable:** Immutable data cannot be changed once created, enhancing predictability in software.
- **Reactive Programming:** Reactive programming involves building asynchronous and event-driven systems.
- **State Management:** State management controls and tracks the state of an application's data over time.
- **Container Registry:** A container registry stores and manages container images for deployment.

- **Dependency Injection:** Dependency injection is a design pattern where dependencies are provided externally to a component.
- **Reusability:** Reusability promotes designing components or code that can be used in multiple contexts.
- **Cross-Platform Development:** Cross-platform development allows creating applications that run on multiple operating systems.
- **Mobile App Development:** Mobile app development involves creating applications for mobile devices like smartphones and tablets.
- **Progressive Web App (PWA):** A PWA is a type of web application that provides an app-like experience with offline capabilities.
- **Augmented Reality (AR):** AR enhances the real-world environment with computer-generated information or experiences.
- **Virtual Reality (VR):** VR creates a simulated environment that immerses users in a virtual world.
- **Edge Computing:** Edge computing processes data closer to the source, reducing latency and improving performance.
- **Deep Learning:** Deep learning is a subset of machine learning using neural networks to model complex patterns.
- **Neural Network:** A neural network is a computer system designed to mimic the human brain's structure and function.
- **Reinforcement Learning:** Reinforcement learning involves training models through rewards or punishments based on actions.
- **Computer Vision:** Computer vision enables machines to interpret and make decisions based on visual data.
- **Natural Language Generation (NLG):** NLG is a process where machines generate human-like language.
- **Feature Engineering:** Feature engineering involves selecting and preparing relevant input variables for machine learning models.
- **Cloud Native:** Cloud-native applications are designed to run in cloud environments, utilizing cloud services.
- **Server Farm:** A server farm is a collection of servers working together to provide services.

- **Immutable Infrastructure:** Immutable infrastructure is a concept where servers are never modified after creation.
- **Service Mesh:** A service mesh manages communication between microservices in a containerized application.
- **Infrastructure Orchestration:** Infrastructure orchestration automates and coordinates the deployment of infrastructure components.
- **Zero Trust Security:** Zero Trust Security assumes no trust, requiring verification from anyone trying to access resources.
- **API Documentation:** API documentation provides information on how to use and integrate with an application programming interface.
- **Swagger:** Swagger is a tool for documenting and testing APIs, facilitating communication between teams.
- **Test Driven Development (TDD):** TDD involves writing tests before developing the actual code to ensure functionality.
- **Behavior Driven Development (BDD):** BDD focuses on collaboration between developers, QA, and non-technical stakeholders in software development.
- **Feature Flag:** A feature flag is a toggle that allows developers to turn a feature on or off remotely.
- **Blue-Green Deployment:** Blue-green deployment involves switching between two identical environments to minimize downtime during updates.
- **Canary Release:** Canary release is a deployment strategy where a new version is gradually rolled out to a subset of users.
- **Dark Launch:** Dark launch is the practice of releasing new features to a subset of users without making them visible.
- **Gitflow:** Gitflow is a branching model that defines a strict branching strategy for Git repositories.
- **Code Freeze:** Code freeze is a period during which no new code is added to a software project to prepare for release.
- **Technical Stack:** A technical stack comprises the technologies and tools used to build a software application.
- **Elasticsearch:** Elasticsearch is a search and analytics engine commonly used for full-text search.
- **Kibana:** Kibana is a data visualization and exploration tool for Elasticsearch.

- **Logstash:** Logstash is a data processing pipeline that ingests, processes, and outputs log data.
- **Fault Tolerance:** Fault tolerance ensures a system remains operational even when components fail.
- **High Availability:** High availability refers to the ability of a system to stay operational with minimal downtime.
- **Disaster Recovery:** Disaster recovery involves strategies and processes for recovering from a catastrophic event.
- **Incident Management:** Incident management addresses and resolves disruptions to IT services.
- **Capacity Management:** Capacity management ensures that IT resources meet current and future business needs.
- **Root Cause Analysis:** Root cause analysis identifies the underlying cause of problems or incidents.
- **Technical Support:** Technical support provides assistance and troubleshooting for users facing issues with software or hardware.
- **Knowledge Base:** A knowledge base is a repository of information used for problem-solving and decision-making.
- **SLA (Service Level Agreement):** SLA defines the level of service expected and the consequences if not met.
- **Micro Frontends:** Micro frontends is an architectural pattern for developing frontend applications in a modular and independent way.
- **Container Registry:** A container registry stores and manages container images for deployment.
- **Microservices Architecture:** Microservices architecture involves developing a system as a collection of small, independent services.
- **Service-Oriented Architecture (SOA):** SOA is an architectural pattern where software components are designed as services that can be accessed remotely.
- **Scalable Architecture:** Scalable architecture allows systems to handle increased workload by adding resources.
- **Legacy System:** A legacy system refers to outdated or obsolete software or hardware.
- **Technical Documentation:** Technical documentation provides information about the design, implementation, and usage of a system.

- **Quality Assurance (QA):** QA involves processes and activities that ensure the quality of software development.
- **Automated Testing:** Automated testing uses tools to execute tests and compare actual outcomes with expected outcomes.
- **Test Case:** A test case is a set of conditions or variables used to determine whether a system functions correctly.
- **Code Repository:** A code repository is a centralized location for storing and managing version-controlled source code.
- **Codebase:** The codebase is the complete body of source code for a software project.
- **Technical Interview:** A technical interview assesses a candidate's knowledge and problem-solving skills in a specific field.
- **Soft Skills:** Soft skills refer to interpersonal, communication, and collaboration abilities.
- **Code Deployment:** Code deployment is the process of releasing a new version of a software application.
- **Rollback:** Rollback is the reversal of a system update or change to a previous state.
- **Continuous Monitoring:** Continuous monitoring tracks the performance and status of systems in real-time.
- **Root Access:** Root access provides full control and privileges over a system.
- **Patch Management:** Patch management involves applying updates and fixes to software to improve security.
- **ISO 27001:** ISO 27001 is an international standard for information security management systems.
- **PCI DSS (Payment Card Industry Data Security Standard):** PCI DSS sets security standards for handling credit card information.
- **Compliance:** Compliance ensures adherence to laws, regulations, and industry standards.
- **GDPR (General Data Protection Regulation):** GDPR is a European regulation on data protection and privacy.
- **HIPAA (Health Insurance Portability and Accountability Act):** HIPAA sets standards for protecting sensitive patient information in healthcare.
- **Sprint Planning:** Sprint planning involves determining the tasks and goals for an upcoming sprint in Agile development.

- **Daily Standup:** Daily standup meetings facilitate communication and coordination in Agile teams.
- **Backlog Grooming:** Backlog grooming involves refining and prioritizing tasks in the backlog for upcoming sprints.
- **Feature Freeze:** Feature freeze is a point in the development cycle where no new features are added to focus on stabilization.
- **User Acceptance Testing (UAT):** UAT involves testing a system with end-users to ensure it meets their requirements.
- **White Box Testing:** White box testing examines the internal logic and structure of a system.
- **Black Box Testing:** Black box testing evaluates a system's functionality without considering internal code.
- **Gray Box Testing:** Gray box testing combines aspects of both white box and black box testing.
- **Mockup:** A mockup is a static representation of a user interface design.
- **Wireframe:** A wireframe is a basic visual representation of a web page or application.
- **Data Lake:** A data lake is a centralized repository for storing raw and unstructured data.
- **Data Mining:** Data mining involves extracting useful patterns and information from large datasets.
- **ETL (Extract, Transform, Load):** ETL processes involve extracting, transforming, and loading data from one system to another.
- **Data Warehouse:** A data warehouse is a centralized repository for storing and managing structured data.
- **Data Mart:** A data mart is a subset of a data warehouse focused on specific business needs.
- **Data Governance:** Data governance ensures data quality, security, and compliance.
- **Data Quality:** Data quality refers to the accuracy, completeness, and consistency of data.
- **Data Privacy:** Data privacy involves protecting sensitive information from unauthorized access.
- **Data Breach:** A data breach is an unauthorized access, disclosure, or acquisition of sensitive information.
- **Data Encryption:** Data encryption secures data by converting it into a coded format.

- **Data Masking:** Data masking involves disguising original data to protect privacy.
- **Data Anonymization:** Data anonymization removes personally identifiable information from datasets.
- **Data Classification:** Data classification categorizes information based on sensitivity and importance.
- **Data Lifecycle:** Data lifecycle management involves managing data from creation to deletion.
- **Data Integration:** Data integration combines and unifies data from different sources for analysis.
- **Data Migration:** Data migration involves transferring data from one system or storage to another.
- **Data Pipeline:** A data pipeline is a set of processes for the automated flow of data from source to destination.
- **Data Replication:** Data replication creates and maintains identical copies of data across multiple systems.
- **Data Science:** Data science involves extracting insights and knowledge from structured and unstructured data.
- **Data Scientist:** A data scientist is a professional who analyzes and interprets complex data sets to inform business decisions.
- **Big Data Analytics:** Big data analytics involves analyzing large and complex datasets to uncover patterns and trends.
- **Data Warehouse:** A data warehouse is a centralized repository for storing and managing structured data.
- **Data Mining:** Data mining involves extracting useful patterns and information from large datasets.
- **Data Cleansing:** Data cleansing is the process of identifying and correcting errors or inconsistencies in datasets.
- **Data Transformation:** Data transformation converts data from one format or structure to another.
- **Data Aggregation:** Data aggregation combines and summarizes information from multiple sources.
- **Data Ingestion:** Data ingestion is the process of collecting and importing data into a system.

- **Data Exploration:** Data exploration involves analyzing and understanding the characteristics of a dataset.
- **Data Visualization:** Data visualization uses graphical representations to convey insights and patterns in data.
- **Data Query:** A data query is a request for specific information from a database or dataset.
- **Data Query Language (DQL):** DQL is a language for retrieving and manipulating data in a database.
- **Data Modeling:** Data modeling is the process of creating a visual representation of the structure and relationships within a dataset.
- **Data Schema:** A data schema defines the structure, organization, and relationships within a database.
- **Data Indexing:** Data indexing enhances database search and retrieval performance by creating a data structure for quick access.
- **Data Backup:** Data backup involves creating copies of data to prevent loss in case of system failure.
- **Data Recovery:** Data recovery is the process of restoring lost or damaged data from backups.
- **Data Archiving:** Data archiving involves moving data to a separate storage for long-term retention.
- **Data Center:** A data center is a facility housing computing systems, servers, and networking equipment.
- **Data Governance:** Data governance ensures data quality, security, and compliance within an organization.
- **Data Stewardship:** Data stewardship involves managing and overseeing the use of data within an organization.
- **Data Ownership:** Data ownership defines responsibility for managing and maintaining specific datasets.
- **Data Access Control:** Data access control restricts access to data based on user roles and permissions.
- **Data Security:** Data security involves protecting data from unauthorized access, disclosure, or alteration.

- **Data Privacy:** Data privacy focuses on protecting individuals' personal information and ensuring legal compliance.
- **Data Compliance:** Data compliance ensures adherence to laws, regulations, and industry standards regarding data handling.
- **Data Ethics:** Data ethics involves considering moral principles and guidelines in the collection and use of data.
- **Data Standards:** Data standards establish uniformity and consistency in data formats and structures.
- **Data Transfer:** Data transfer involves moving data from one location or system to another.
- **Data Stream:** A data stream is a continuous flow of data generated in real-time.
- **Data Lake:** A data lake is a centralized repository for storing raw and unstructured data.
- **Data Warehousing:** Data warehousing involves collecting, storing, and managing data from different sources for analysis.
- **Data Mart:** A data mart is a subset of a data warehouse focused on specific business needs.
- **Data Quality:** Data quality refers to the accuracy, completeness, and consistency of data.
- **Data Profiling:** Data profiling assesses the quality and characteristics of data in a dataset.
- **Data Lineage:** Data lineage tracks the origin, movement, and changes to data throughout its lifecycle.
- **Data Validation:** Data validation ensures that data meets specified criteria and standards.
- **Data Correlation:** Data correlation identifies relationships and dependencies between different datasets.
- **Data Enrichment:** Data enrichment involves enhancing existing data with additional information or attributes.
- **Data Retention:** Data retention defines how long data should be stored and when it should be deleted.
- **Data Disposal:** Data disposal involves securely and permanently removing data that is no longer needed.

- **Data Ecosystem:** A data ecosystem encompasses all the components, tools, and processes involved in managing and utilizing data within an organization.
- **Data Strategy:** Data strategy outlines the plan and objectives for effectively managing, processing, and utilizing data to achieve organizational goals.
- **Data Governance Framework:** A data governance framework provides the structure and guidelines for ensuring data quality, security, and compliance.
- **Data Warehouse Architecture:** Data warehouse architecture defines the structure and design of a centralized repository for structured data.
- **Data Lake Architecture:** Data lake architecture outlines the design and organization of a centralized repository for raw and unstructured data.
- **Webhooks:** Webhooks are automated callbacks or HTTP POST requests triggered by events, allowing real-time data exchange between systems.
- **JWT (JSON Web Token):** JWT is a compact, URL-safe means of representing claims to be transferred between parties, commonly used for authentication.
- **OAuth (Open Authorization):** OAuth is an open standard for authorization, allowing secure access to resources without sharing credentials.
- **PKI (Public Key Infrastructure):** PKI is a framework that manages digital keys and certificates to ensure secure communication.
- **SSL/TLS (Secure Sockets Layer/Transport Layer Security):** SSL/TLS are cryptographic protocols ensuring secure communication over a computer network.
- **HTTP/HTTPS:** HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are protocols for transmitting data between a web server and a browser.
- **Load Balancer:** A load balancer distributes incoming network traffic across multiple servers to ensure optimal resource utilization and prevent server overload.
- **Reverse Proxy:** A reverse proxy handles requests from clients and forwards them to servers, enhancing security and performance.
- **CDN (Content Delivery Network):** A CDN is a distributed network of servers that deliver web content to users based on their geographic location, improving speed and reliability.
- **Server Logs:** Server logs record events and activities on a server, providing valuable information for troubleshooting and analysis.

- **Monitoring:** Monitoring involves observing and tracking the performance and health of systems in real-time.
- **Logging:** Logging records events, activities, and errors for analysis, troubleshooting, and auditing.
- **Alerting:** Alerting notifies relevant parties of critical events or issues requiring immediate attention.
- **Incident Response:** Incident response is a structured approach to addressing and managing cybersecurity incidents.
- **Root Cause Analysis:** Root cause analysis identifies the underlying cause of problems or incidents.
- **Performance Metrics:** Performance metrics measure the efficiency and effectiveness of a system or process.
- **Scalability:** Scalability is the ability of a system to handle increased workload or demand by adding resources.
- **High Availability:** High availability ensures a system remains operational with minimal downtime.
- **Fault Tolerance:** Fault tolerance ensures a system can continue to operate even when components fail.
- **Latency:** Latency is the time delay between a request and the corresponding response in a system.
- **Throughput:** Throughput is the amount of data or transactions a system can process within a given time.
- **Redundancy:** Redundancy involves duplicating critical components to ensure system reliability.
- **Failover:** Failover is the process of switching to a backup system in case the primary system fails.
- **Disaster Recovery:** Disaster recovery involves strategies and processes for recovering from catastrophic events.
- **Backup and Restore:** Backup and restore are procedures for creating copies of data and restoring it if necessary.
- **Business Continuity:** Business continuity planning ensures an organization can continue essential functions during and after a disaster.

- **Change Management:** Change management controls and manages changes to a system or process to minimize disruptions.
- **Patch Management:** Patch management involves applying updates and fixes to software for security and performance.
- **Vulnerability Management:** Vulnerability management identifies and addresses weaknesses in a system's security.
- **Penetration Testing:** Penetration testing assesses the security of a system by simulating a cyberattack.
- **Firewall:** A firewall is a network security device that monitors and controls incoming and outgoing network traffic.
- **Intrusion Detection System (IDS):** An IDS monitors and detects potential security threats or breaches.
- **Security Audit:** A security audit assesses and evaluates the effectiveness of security controls and policies.
- **Zero Trust Security:** Zero Trust Security assumes no trust, requiring verification from anyone trying to access resources.
- **Encryption:** Encryption secures data by converting it into a coded format that can only be deciphered with the appropriate key.
- **Access Control:** Access control restricts access to resources based on user roles and permissions.
- **Multi-Factor Authentication (MFA):** MFA adds an extra layer of security by requiring multiple methods of authentication.
- **Security Policy:** A security policy outlines rules and guidelines to ensure the security of an organization's information assets.
- **Security Compliance:** Security compliance ensures adherence to laws, regulations, and industry standards regarding information security.
- **DDoS Mitigation:** DDoS mitigation involves strategies to prevent or minimize the impact of distributed denial-of-service attacks.
- **Security Incident:** A security incident refers to any event that poses a risk to the confidentiality, integrity, or availability of information.
- **Phishing:** Phishing is a cyber attack where attackers attempt to trick individuals into revealing sensitive information through deceptive emails or websites.

- **Malware:** Malware, short for malicious software, is software designed to harm or exploit computer systems, including viruses, worms, and trojans.
- **Ransomware:** Ransomware is a type of malware that encrypts a user's data and demands a ransom for its release.
- **Security Awareness:** Security awareness involves educating individuals about potential security threats and promoting safe online practices.
- **Cybersecurity Framework:** A cybersecurity framework provides a structured approach to managing and improving an organization's cybersecurity posture.
- **SOC (Security Operations Center):** A Security Operations Center is a centralized unit that monitors and manages an organization's security infrastructure.
- **Threat Intelligence:** Threat intelligence involves analyzing and understanding potential cybersecurity threats to enhance security measures.
- **Security Patch:** A security patch is a software update that addresses vulnerabilities and improves the security of a system or application.
- **Secure Coding:** Secure coding is the practice of writing code in a way that mitigates security risks and vulnerabilities.
- **Web Application Firewall (WAF):** A Web Application Firewall is a security device or service that protects web applications from various cyber threats.
- **Mobile Device Management (MDM):** Mobile Device Management involves managing and securing mobile devices within an organization.
- **Network Segmentation:** Network segmentation divides a network into segments to enhance security by restricting access between segments.

SECOND PART: test your comprehension

Listen the vocabulary alone to remember the meaning and repeat each word

- Software Engineering
- Programming Language
- Algorithm
- Code

- Debugging
- Version Control
- Database
- Data Structure
- API (Application Programming Interface)
- Backend
- Frontend
- User Interface (UI)
- User Experience (UX)
- Framework
- Integration Testing
- Deployment
- Agile
- Scrum
- DevOps
- Artificial Intelligence (AI)
- Machine Learning
- Cybersecurity
- Cloud Computing
- Big Data
- Virtualization
- Networking
- Debug
- Patch
- Encryption
- Open Source
- Scalability
- Code Review
- Repository
- Continuous Integration (CI)
- Continuous Deployment (CD)
- Full Stack

- Responsive Design
- Backend as a Service (BaaS)
- Frontend Framework
- User Authentication
- Framework
- Library
- Dependency Injection
- Runtime
- Serverless
- Microservices
- Containerization
- Docker
- Kubernetes
- Scalability
- Load Balancing
- Failover
- Concurrency
- Parallelism
- Multithreading
- Multiprocessing
- API Endpoint
- RESTful
- SOAP
- JSON (JavaScript Object Notation)
- XML (eXtensible Markup Language)
- Regular Expression
- Dependency Management
- CI/CD Pipeline
- Unit Testing
- Integration Testing
- End-to-End Testing
- Refactoring

- Code Smell
- Legacy Code
- Technical Debt
- Codebase
- Code Repository
- Git
- Mercurial
- Continuous Monitoring
- Logging
- Metrics
- Performance Tuning
- Profiling
- Code Review
- Pair Programming
- Agile Manifesto
- Sprint
- Backlog
- Burn Down Chart
- JIRA
- Confluence
- Epic
- User Story
- Wireframe
- Prototyping
- MVP (Minimum Viable Product)
- AB Testing
- DevSecOps
- Cryptography
- Blockchain
- Smart Contract
- Frontend Framework
- Backend Framework

- Static Analysis
- Dynamic Analysis
- Load Testing
- Stress Testing
- Capacity Planning
- Caching
- Latency
- WebSockets
- REST API
- GraphQL
- OAuth
- JWT (JSON Web Token)
- Single Sign-On (SSO)
- Two-Factor Authentication (2FA)
- Cross-Origin Resource Sharing (CORS)
- OWASP
- Zero-Day Vulnerability
- Penetration Testing
- Incident Response
- Data Migration
- ETL (Extract, Transform, Load)
- Data Warehousing
- Business Intelligence (BI)
- DevOps Tools
- Container Orchestration
- Infrastructure as Code (IaC)
- Server Farm
- VPN (Virtual Private Network)
- Firewall
- Intrusion Detection System (IDS)
- Distributed Systems
- Micro Frontends

- Chaos Engineering
- Dark Mode
- Chatbot
- Natural Language Processing (NLP)
- Predictive Analytics
- Robotic Process Automation (RPA)
- Serverless Computing
- Low-Code Development
- API Gateway
- Server-Side Rendering (SSR)
- Client-Side Rendering (CSR)
- Immutable
- Reactive Programming
- State Management
- Container Registry
- Dependency Injection
- Reusability
- Cross-Platform Development
- Mobile App Development
- Progressive Web App (PWA)
- Augmented Reality (AR)
- Virtual Reality (VR)
- Edge Computing
- Deep Learning
- Neural Network
- Reinforcement Learning
- Computer Vision
- Natural Language Generation (NLG)
- Feature Engineering
- Cloud Native
- Server Farm
- Immutable Infrastructure

- Service Mesh
- Infrastructure Orchestration
- Zero Trust Security
- API Documentation
- Swagger
- Test Driven Development (TDD)
- Behavior Driven Development (BDD)
- Feature Flag
- Blue-Green Deployment
- Canary Release
- Dark Launch
- Gitflow
- Code Freeze
- Technical Stack
- Elasticsearch
- Kibana
- Logstash
- Fault Tolerance
- High Availability
- Disaster Recovery
- Incident Management
- Capacity Management
- Root Cause Analysis
- Technical Support
- Knowledge Base
- SLA (Service Level Agreement)
- Micro Frontends
- Container Registry
- Microservices Architecture
- Service-Oriented Architecture (SOA)
- Scalable Architecture
- Legacy System

- Technical Documentation
- Quality Assurance (QA)
- Automated Testing
- Test Case
- Code Repository
- Codebase
- Technical Interview
- Soft Skills
- Code Deployment
- Rollback
- Continuous Monitoring
- Root Access
- Patch Management
- ISO 27001
- PCI DSS (Payment Card Industry Data Security Standard)
- Compliance
- GDPR (General Data Protection Regulation)
- HIPAA (Health Insurance Portability and Accountability Act)
- Sprint Planning
- Daily Standup
- Backlog Grooming
- Feature Freeze
- User Acceptance Testing (UAT)
- White Box Testing
- Black Box Testing
- Gray Box Testing
- Mockup
- Wireframe
- Data Lake
- Data Mining
- ETL (Extract, Transform, Load)
- Data Warehouse

- Data Mart
- Data Governance
- Data Quality
- Data Privacy
- Data Breach
- Data Encryption
- Data Masking
- Data Anonymization
- Data Classification
- Data Lifecycle
- Data Integration
- Data Migration
- Data Pipeline
- Data Replication
- Data Science
- Data Scientist
- Big Data Analytics
- Data Warehouse
- Data Mining
- Data Cleansing
- Data Transformation
- Data Aggregation
- Data Ingestion
- Data Exploration
- Data Visualization
- Data Query
- Data Query Language (DQL)
- Data Modeling
- Data Schema
- Data Indexing
- Data Backup
- Data Recovery

- Data Archiving
- Data Center
- Data Governance
- Data Stewardship
- Data Ownership
- Data Access Control
- Data Security
- Data Privacy
- Data Compliance
- Data Ethics
- Data Standards
- Data Transfer
- Data Stream
- Data Lake
- Data Warehousing
- Data Mart
- Data Quality
- Data Profiling
- Data Lineage
- Data Validation
- Data Correlation
- Data Enrichment
- Data Retention
- Data Disposal
- Data Ecosystem
- Data Strategy
- Data Governance Framework
- Data Warehouse Architecture
- Data Lake Architecture
- Webhooks
- JWT (JSON Web Token)
- OAuth (Open Authorization)

- PKI (Public Key Infrastructure)
- SSL/TLS (Secure Sockets Layer/Transport Layer Security)
- HTTP/HTTPS
- Load Balancer
- Reverse Proxy
- CDN (Content Delivery Network)
- Server Logs
- Monitoring
- Logging
- Alerting
- Incident Response
- Root Cause Analysis
- Performance Metrics
- Scalability
- High Availability
- Fault Tolerance
- Latency
- Throughput
- Redundancy
- Failover
- Disaster Recovery
- Backup and Restore
- Business Continuity
- Change Management
- Patch Management
- Vulnerability Management
- Penetration Testing
- Firewall
- Intrusion Detection System (IDS)
- Security Audit
- Zero Trust Security
- Encryption

- Access Control
- Multi-Factor Authentication (MFA)
- Security Policy
- Security Compliance
- DDoS Mitigation
- Security Incident
- Phishing
- Malware
- Ransomware
- Security Awareness
- Cybersecurity Framework
- SOC (Security Operations Center)
- Threat Intelligence
- Security Patch
- Secure Coding
- Web Application Firewall (WAF)
- Mobile Device Management (MDM)
- Network Segmentation
-

TRANSLATION

- **Software Engineering** : Ingénierie logicielle
- **Programming Language** : Langage de programmation
- **Algorithm** : Algorithme
- **Code** : Code
- **Debugging** : Débogage

- **Version Control** : Contrôle de version
- **Database** : Base de données
- **Data Structure** : Structure de données
- **API (Application Programming Interface)** : Interface de programmation d'application
- **Backend** : Partie serveur (côté serveur)
- **Frontend** : Partie client (côté client)
- **User Interface (UI)** : Interface utilisateur
- **User Experience (UX)** : Expérience utilisateur
- **Framework** : Cadre de travail
- **Integration Testing** : Test d'intégration
- **Deployment** : Déploiement
- **Agile** : Agile
- **Scrum** : Scrum
- **DevOps** : DevOps
- **Artificial Intelligence (AI)** : Intelligence artificielle
- **Machine Learning** : Apprentissage automatique
- **Cybersecurity** : Cybersécurité
- **Cloud Computing** : Informatique en nuage
- **Big Data** : Big Data
- **Virtualization** : Virtualisation
- **Networking** : Réseautage
- **Debug** : Déboguer
- **Patch** : Correctif
- **Encryption** : Chiffrement
- **Open Source** : Open source
- **Scalability** : Scalabilité
- **Code Review** : Revue de code
- **Repository** : Dépôt
- **Continuous Integration (CI)** : Intégration continue
- **Continuous Deployment (CD)** : Déploiement continu
- **Full Stack** : Développeur full stack
- **Responsive Design** : Conception adaptative

- **Backend as a Service (BaaS)** : Backend en tant que service
- **Frontend Framework** : Cadre frontal
- **User Authentication** : Authentification utilisateur
- **Framework** : Cadre de travail
- **Library** : Bibliothèque
- **Dependency Injection** : Injection de dépendances
- **Runtime** : Exécution
- **Serverless** : Sans serveur
- **Microservices** : Microservices
- **Containerization** : Conteneurisation
- **Docker** : Docker
- **Kubernetes** : Kubernetes
- **Scalability** : Scalabilité
- **Load Balancing** : Équilibrage de charge
- **Failover** : Basculement
- **Concurrency** : Concurrence
- **Parallelism** : Parallélisme
- **Multithreading** : Multithreading
- **Multiprocessing** : Multiprocessing
- **API Endpoint** : Point d'accès de l'API
- **RESTful** : RESTful
- **SOAP** : SOAP
- **JSON (JavaScript Object Notation)** : JSON (Notation d'objet JavaScript)
- **XML (eXtensible Markup Language)** : XML (Langage de balisage extensible)
- **Regular Expression** : Expression régulière
- **Dependency Management** : Gestion des dépendances
- **CI/CD Pipeline** : Pipeline CI/CD
- **Unit Testing** : Test unitaire
- **Integration Testing** : Test d'intégration
- **End-to-End Testing** : Test de bout en bout
- **Refactoring** : Refactoring
- **Code Smell** : Code malodorant (indique un problème de conception)

- **Legacy Code** : Code hérité (ancien code)
- **Technical Debt** : Dette technique
- **Codebase** : Base de code
- **Code Repository** : Répertoire de code
- **Git** : Git
- **Mercurial** : Mercurial
- **Continuous Monitoring** : Surveillance continue
- **Logging** : Journalisation
- **Metrics** : Métriques
- **Performance Tuning** : Optimisation des performances
- **Profiling** : Profilage
- **Code Review** : Revue de code
- **Pair Programming** : Programmation en binôme
- **Agile Manifesto** : Manifeste Agile
- **Sprint** : Sprint
- **Backlog** : Liste des tâches
- **Burn Down Chart** : Diagramme de burn down
- **JIRA** : JIRA (outil de suivi de projet)
- **Confluence** : Confluence (outil de collaboration)
- **Epic** : Épique (groupe de tâches)
- **User Story** : Histoire utilisateur
- **Wireframe** : Maquette filaire
- **Prototyping** : Prototypage
- **MVP (Minimum Viable Product)** : Produit minimum viable
- **AB Testing** : Test A/B
- **DevSecOps** : DevSecOps (intégration de la sécurité dans le développement et l'exploitation)
- **Cryptography** : Cryptographie
- **Blockchain** : Blockchain
- **Smart Contract** : Contrat intelligent
- **Frontend Framework** : Cadre frontal
- **Backend Framework** : Cadre arrière

- **Static Analysis** : Analyse statique
- **Dynamic Analysis** : Analyse dynamique
- **Load Testing** : Test de charge
- **Stress Testing** : Test de stress
- **Capacity Planning** : Planification de capacité
- **Caching** : Mise en cache
- **Latency** : Latence
- **WebSockets** : WebSockets
- **REST API** : API REST
- **GraphQL** : GraphQL
- **OAuth** : OAuth
- **JWT (JSON Web Token)** : JWT (Jeton web JSON)
- **Single Sign-On (SSO)** : Authentification unique
- **Two-Factor Authentication (2FA)** : Authentification à deux facteurs
- **Cross-Origin Resource Sharing (CORS)** : Partage des ressources entre origines différentes
- **OWASP** : OWASP (Open Web Application Security Project)
- **Zero-Day Vulnerability** : Vulnérabilité zero-day
- **Penetration Testing** : Test d'intrusion
- **Incident Response** : Gestion des incidents
- **Data Migration** : Migration de données
- **ETL (Extract, Transform, Load)** : ETL (Extraction, Transformation, Chargement)
- **Data Warehousing** : Entrepôt de données
- **Business Intelligence (BI)** : Intelligence d'affaires
- **DevOps Tools** : Outils DevOps
- **Container Orchestration** : Orchestration de conteneurs
- **Infrastructure as Code (IaC)** : Infrastructure en tant que code
- **Server Farm** : Ferme de serveurs
- **VPN (Virtual Private Network)** : VPN (Réseau privé virtuel)
- **Firewall** : Pare-feu
- **Intrusion Detection System (IDS)** : Système de détection d'intrusion
- **Distributed Systems** : Systèmes distribués

- **Micro Frontends** : Micro frontends
- **Chaos Engineering** : Ingénierie du chaos
- **Dark Mode** : Mode sombre
- **Chatbot** : Chatbot
- **Natural Language Processing (NLP)** : Traitement du langage naturel
- **Predictive Analytics** : Analyse prédictive
- **Robotic Process Automation (RPA)** : Automatisation des processus robotiques
- **Serverless Computing** : Informatique sans serveur
- **Low-Code Development** : Développement à faible code
- **API Gateway** : Passerelle API
- **Server-Side Rendering (SSR)** : Rendu côté serveur
- **Client-Side Rendering (CSR)** : Rendu côté client
- **Immutable** : Immuable
- **Reactive Programming** : Programmation réactive
- **State Management** : Gestion de l'état
- **Container Registry** : Registre de conteneurs
- **Dependency Injection** : Injection de dépendances
- **Reusability** : Réutilisabilité
- **Cross-Platform Development** : Développement multiplateforme
- **Mobile App Development** : Développement d'application mobile
- **Progressive Web App (PWA)** : Application web progressive
- **Augmented Reality (AR)** : Réalité augmentée
- **Virtual Reality (VR)** : Réalité virtuelle
- **Edge Computing** : Informatique en périphérie
- **Deep Learning** : Apprentissage profond
- **Neural Network** : Réseau neuronal
- **Reinforcement Learning** : Apprentissage par renforcement
- **Computer Vision** : Vision par ordinateur
- **Natural Language Generation (NLG)** : Génération de langage naturel
- **Feature Engineering** : Ingénierie des caractéristiques
- **Cloud Native** : Natif du cloud
- **Server Farm** : Ferme de serveurs

- **Immutable Infrastructure** : Infrastructure immuable
- **Service Mesh** : Maillage de services
- **Infrastructure Orchestration** : Orchestration d'infrastructure
- **Zero Trust Security** : Sécurité de confiance zéro
- **API Documentation** : Documentation de l'API
- **Swagger** : Swagger (outil de documentation API)
- **Test Driven Development (TDD)** : Développement piloté par les tests
- **Behavior Driven Development (BDD)** : Développement piloté par le comportement
- **Feature Flag** : Drapeau de fonction
- **Blue-Green Deployment** : Déploiement bleu-vert
- **Canary Release** : Version canari (libération progressive)
- **Dark Launch** : Lancement sombre
- **Gitflow** : Gitflow (stratégie de gestion de branches Git)
- **Code Freeze** : Gel du code
- **Technical Stack** : Pile technologique
- **Elasticsearch** : Elasticsearch (moteur de recherche et d'analyse de données)
- **Kibana** : Kibana (outil de visualisation de données pour Elasticsearch)
- **Logstash** : Logstash (outil de collecte, traitement et expédition de logs)
- **Fault Tolerance** : Tolérance aux pannes
- **High Availability** : Haute disponibilité
- **Disaster Recovery** : Récupération après sinistre
- **Incident Management** : Gestion des incidents
- **Capacity Management** : Gestion de la capacité
- **Root Cause Analysis** : Analyse des causes premières
- **Technical Support** : Support technique
- **Knowledge Base** : Base de connaissances
- **SLA (Service Level Agreement)** : Accord de niveau de service
- **Micro Frontends** : Micro frontends
- **Container Registry** : Registre de conteneurs
- **Microservices Architecture** : Architecture microservices
- **Service-Oriented Architecture (SOA)** : Architecture orientée services (AOS)
- **Scalable Architecture** : Architecture évolutive

- **Legacy System** : Système hérité
- **Technical Documentation** : Documentation technique
- **Quality Assurance (QA)** : Assurance qualité
- **Automated Testing** : Test automatisé
- **Test Case** : Cas de test
- **Code Repository** : Répertoire de code
- **Codebase** : Base de code
- **Technical Interview** : Entretien technique
- **Soft Skills** : Compétences sociales
- **Code Deployment** : Déploiement de code
- **Rollback** : Retour en arrière
- **Continuous Monitoring** : Surveillance continue
- **Root Access** : Accès root
- **Patch Management** : Gestion des correctifs
- **Vulnerability Assessment** : Évaluation des vulnérabilités
- **ISO 27001** : ISO 27001 (norme de sécurité de l'information)
- **PCI DSS (Payment Card Industry Data Security Standard)** : PCI DSS (norme de sécurité des données de l'industrie des cartes de paiement)
- **Compliance** : Conformité
- **GDPR (General Data Protection Regulation)** : GDPR (Règlement général sur la protection des données)
- **HIPAA (Health Insurance Portability and Accountability Act)** : HIPAA (loi américaine sur la portabilité et la responsabilité en matière d'assurance maladie)
- **Sprint Planning** : Planification de sprint
- **Daily Standup** : Réunion quotidienne debout
- **Backlog Grooming** : Préparation du backlog
- **Feature Freeze** : Gel des fonctionnalités
- **User Acceptance Testing (UAT)** : Test d'acceptation utilisateur
- **White Box Testing** : Test en boîte blanche
- **Black Box Testing** : Test en boîte noire
- **Gray Box Testing** : Test en boîte grise
- **Mockup** : Maquette

- **Wireframe** : Maquette filaire
- **Data Lake** : Lac de données
- **Data Mining** : Exploration de données
- **ETL (Extract, Transform, Load)** : ETL (Extraction, Transformation, Chargement)
- **Data Warehouse** : Entrepôt de données
- **Data Mart** : Magasin de données
- **Data Governance** : Gouvernance des données
- **Data Quality** : Qualité des données
- **Data Privacy** : Confidentialité des données
- **Data Breach** : Violation de données
- **Data Encryption** : Chiffrement des données
- **Data Masking** : Masquage des données
- **Data Anonymization** : Anonymisation des données
- **Data Classification** : Classification des données
- **Data Lifecycle** : Cycle de vie des données
- **Data Integration** : Intégration des données
- **Data Migration** : Migration de données
- **Data Pipeline** : Pipeline de données
- **Data Replication** : Réplication des données
- **Data Science** : Science des données
- **Data Scientist** : Scientifique des données
- **Big Data Analytics** : Analyse de big data
- **Data Warehouse** : Entrepôt de données
- **Data Mining** : Exploration de données
- **Data Cleansing** : Nettoyage des données
- **Data Transformation** : Transformation des données
- **Data Aggregation** : Agrégation des données
- **Data Ingestion** : Ingestion des données
- **Data Exploration** : Exploration des données
- **Data Visualization** : Visualisation des données
- **Data Query** : Requête de données
- **Data Query Language (DQL)** : Langage de requête de données

- **Data Modeling** : Modélisation des données
- **Data Schema** : Schéma de données
- **Data Indexing** : Indexation des données
- **Data Backup** : Sauvegarde des données
- **Data Recovery** : Récupération des données
- **Data Archiving** : Archivage des données
- **Data Center** : Centre de données
- **Data Governance** : Gouvernance des données
- **Data Stewardship** : Gestion des données
- **Data Ownership** : Propriété des données
- **Data Access Control** : Contrôle d'accès aux données
- **Data Security** : Sécurité des données
- **Data Privacy** : Confidentialité des données
- **Data Compliance** : Conformité des données
- **Data Ethics** : Éthique des données
- **Data Standards** : Normes de données
- **Data Transfer** : Transfert de données
- **Data Stream** : Flux de données
- **Data Lake** : Lac de données
- **Data Warehousing** : Entrepôt de données
- **Data Mart** : Magasin de données
- **Data Quality** : Qualité des données
- **Data Profiling** : Profilage des données
- **Data Lineage** : Lignage des données
- **Data Validation** : Validation des données
- **Data Correlation** : Corrélation des données
- **Data Enrichment** : Enrichissement des données
- **Data Retention** : Conservation des données
- **Data Disposal** : Élimination des données
- **Data Ecosystem** : Écosystème de données
- **Data Strategy** : Stratégie de données
- **Data Governance Framework** : Cadre de gouvernance des données

- **Data Warehouse Architecture** : Architecture d'entrepôt de données
- **Data Lake Architecture** : Architecture de lac de données
- **Webhooks** : Webhooks
- **JWT (JSON Web Token)** : JWT (Jeton Web JSON)
- **OAuth (Open Authorization)** : OAuth (Autorisation ouverte)
- **PKI (Public Key Infrastructure)** : Infrastructure à clés publiques
- **SSL/TLS (Secure Sockets Layer/Transport Layer Security)** : SSL/TLS (Protocole sécurisé de la couche transport)
- **HTTP/HTTPS** : HTTP/HTTPS
- **Load Balancer** : Équilibreur de charge
- **Reverse Proxy** : Proxy inverse
- **CDN (Content Delivery Network)** : CDN (Réseau de diffusion de contenu)
- **Server Logs** : Journaux serveur
- **Monitoring** : Surveillance
- **Logging** : Journalisation
- **Alerting** : Alerte
- **Incident Response** : Gestion des incidents
- **Root Cause Analysis** : Analyse des causes premières
- **Performance Metrics** : Métriques de performance
- **Scalability** : Scalabilité
- **High Availability** : Haute disponibilité
- **Fault Tolerance** : Tolérance aux pannes
- **Latency** : Latence
- **Throughput** : Débit
- **Redundancy** : Redondance
- **Failover** : Basculement
- **Disaster Recovery** : Récupération après sinistre
- **Backup and Restore** : Sauvegarde et restauration
- **Business Continuity** : Continuité des activités
- **Change Management** : Gestion du changement
- **Patch Management** : Gestion des correctifs
- **Vulnerability Management** : Gestion des vulnérabilités

- **Penetration Testing** : Test d'intrusion
- **Firewall** : Pare-feu
- **Intrusion Detection System (IDS)** : Système de détection d'intrusion
- **Security Audit** : Audit de sécurité
- **Zero Trust Security** : Sécurité de confiance zéro
- **Encryption** : Chiffrement
- **Access Control** : Contrôle d'accès
- **Multi-Factor Authentication (MFA)** : Authentification à plusieurs facteurs
- **Security Policy** : Politique de sécurité
- **Security Compliance** : Conformité en matière de sécurité
- **DDoS Mitigation** : Atténuation des attaques par déni de service (DDoS)
- **Security Incident** : Incident de sécurité
- **Phishing** : Hameçonnage
- **Malware** : Logiciel malveillant
- **Ransomware** : Rançongiciel
- **Security Awareness** : Sensibilisation à la sécurité
- **Cybersecurity Framework** : Cadre de cybersécurité
- **SOC (Security Operations Center)** : Centre des opérations de sécurité
- **Threat Intelligence** : Intelligence des menaces
- **Security Patch** : Correctif de sécurité
- **Secure Coding** : Codage sécurisé
- **Web Application Firewall (WAF)** : Pare-feu d'application web
- **Mobile Device Management (MDM)** : Gestion des appareils mobiles
- **Network Segmentation** : Segmentation du réseau