

Artificial Intelligence Tutorial | AI Tutorial

-
-
-

Artificial Intelligence Tutorial | AI Tutorial

Artificial Intelligence(AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. It involves the development of algorithms and computer programs that can perform tasks that typically require human intelligence such as visual perception, speech recognition, decision-making, and language translation.

Artificial Intelligence Tutorial | AI Tutorial

There are various Definition provided by the scientists of various fields about Artificial Intelligence, some of them are mentioned below:

Artificial Intelligence Tutorial | AI Tutorial

"Artificial Intelligence is the study of how to make computers do things at which, at the movement, people are better". ~Rich and Knight (1991)

Artificial Intelligence Tutorial | AI Tutorial

"Artificial Intelligence is the study of the computations that make it possible to perceive, reason and act." ~Winston (1992)

Artificial Intelligence Tutorial | AI Tutorial

"AI is the study of mental faculties through the use of computational models". ~Charniak and McDermott (1985)

Types of Artificial Intelligence

Artificial Intelligence (AI) is broadly classified into:

Types of Artificial Intelligence

- Types of AI Based on Capabilities
- Types of AI Based on Functionalities

What is an AI Agent?

An AI agent is a software or hardware entity that performs actions autonomously, with the goal of achieving specific objectives. It operates by perceiving its environment, processing information, making decisions, and taking actions based on its perceptions and goals.

Types of AI Agents

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agents
- Multi-Agent Systems

Problem Solving in AI

Problem-solving is a fundamental aspect of AI, involving the design and application of algorithms to solve complex problems systematically. AI systems utilize various problem-solving techniques to find solutions efficiently and effectively.

1. Search Algorithms in AI

Search algorithms navigate through problem spaces to find solutions. They can be categorized into uninformed search and informed searches.

1. Search Algorithms in AI

- Uninformed Search Algorithm explores the search space without any domain-specific knowledge beyond the problem's definition. These algorithms do not use any additional information like

heuristics to guide the search. Breadth-First Search (BFS) Depth-First Search (DFS) Uniform Cost Search (UCS) Iterative Deepening Search Bidirectional search

- Breadth-First Search (BFS)
- Depth-First Search (DFS)
- Uniform Cost Search (UCS)
- Iterative Deepening Search
- Bidirectional search
- Informed Search Algorithm use additional information (heuristics) to make decisions about which paths to explore. This helps in efficiently finding solutions by guiding the search process towards more promising paths. Greedy Best-First Search A Search* Algorithm Simplified Memory-Bounded A* (SMA*)
- Greedy Best-First Search
- A Search* Algorithm
- Simplified Memory-Bounded A* (SMA*)

1. Search Algorithms in AI

- Breadth-First Search (BFS)
- Depth-First Search (DFS)
- Uniform Cost Search (UCS)
- Iterative Deepening Search
- Bidirectional search

1. Search Algorithms in AI

- Greedy Best-First Search
- A Search* Algorithm
- Simplified Memory-Bounded A* (SMA*)

2. Local Search Algorithms

Local search algorithms operate on a single current state (or a small set of states) and attempt to improve it incrementally by exploring neighboring states.

2. Local Search Algorithms

- Hill-Climbing Search Algorithm
- Simulated Annealing
- Local Beam Search
- Genetic Algorithms
- Tabu Search

3. Adversarial Search in AI

Adversarial search deals with competitive environments where multiple agents (often two) are in direct competition with one another, such as in games like chess, tic-tac-toe, or Go.

3. Adversarial Search in AI

- Minimax Algorithm
- Alpha-Beta Pruning
- Expectiminimax Algorithm

4. Constraint Satisfaction Problems

Constraint Satisfaction Problem (CSP) is a problem-solving framework that involves variables, each with a domain of possible values, and constraints limiting the combinations of variable values. The objective is to find a consistent assignment satisfying all constraints.

4. Constraint Satisfaction Problems

- Constraint Propagation in CSPs
- Backtracking Search for CSPs

Knowledge, Reasoning and Planning in AI

Knowledge representation in Artificial Intelligence (AI) refers to the way information, knowledge, and

data are structured, stored, and used by AI systems to reason, learn, and make decisions. Common techniques for knowledge representation include:

Knowledge, Reasoning and Planning in AI

- Semantic Networks
- Frames
- Ontologies
- Logical Representation
- Production Rules

First Order Logic in Artificial Intelligence

First Order Logic (FOL) is used to represent knowledge and reason about the world. FOL allows for the expression of more complex statements involving objects, their properties, and the relationships between them.

First Order Logic in Artificial Intelligence

- Knowledge Representation in First Order Logic
- Syntax and Semantics of First Order Logic
- Inference Rules in First Order Logic

Reasoning in Artificial Intelligence

Reasoning in Artificial Intelligence (AI) is the process by which AI systems draw conclusions, make decisions, or infer new knowledge from existing information. Types of reasoning used in AI are:

Reasoning in Artificial Intelligence

- Deductive Reasoning
- Inductive Reasoning
- Abductive Reasoning
- Fuzzy Reasoning

Reasoning in Artificial Intelligence

To learn more about reasoning in AI, you can refer to: [Types of Reasoning in AI](#)

Planning in AI

Planning in AI generates a sequence of actions that an intelligent agent needs to execute to achieve specific goals or objectives. Some of the planning techniques in artificial intelligence includes:

Planning in AI

- Classical Planning: Assumes a deterministic environment where actions have predictable outcomes. STRIPS (Stanford Research Institute Problem Solver) PDDL (Planning Domain Definition Language) Forward State Space Search
- STRIPS (Stanford Research Institute Problem Solver)
- PDDL (Planning Domain Definition Language)
- Forward State Space Search
- Probabilistic Planning: Deals with uncertainty in the environment, where actions may have probabilistic outcomes. Markov Decision Processes (MDPs) Partially Observable Markov Decision Processes (POMDPs) Monte Carlo Tree Search (MCTS)
- Markov Decision Processes (MDPs)
- Partially Observable Markov Decision Processes (POMDPs)
- Monte Carlo Tree Search (MCTS)
- Hierarchical Planning: Breaks down complex tasks into simpler sub-tasks, often using a hierarchy of plans to solve different levels of the problem. Hierarchical Task Networks (HTNs) Hierarchical Reinforcement Learning (HRL) Hierarchical State Space Search (HSSS)
- Hierarchical Task Networks (HTNs)
- Hierarchical Reinforcement Learning (HRL)
- Hierarchical State Space Search (HSSS)

Planning in AI

- STRIPS (Stanford Research Institute Problem Solver)
- PDDL (Planning Domain Definition Language)
- Forward State Space Search

Planning in AI

- Markov Decision Processes (MDPs)
- Partially Observable Markov Decision Processes (POMDPs)
- Monte Carlo Tree Search (MCTS)

Planning in AI

- Hierarchical Task Networks (HTNs)
- Hierarchical Reinforcement Learning (HRL)
- Hierarchical State Space Search (HSSS)

Uncertain Knowledge and Reasoning in AI

Uncertain Knowledge and Reasoning in AI refers to the methods and techniques used to handle situations where information is incomplete, ambiguous, or uncertain. For managing uncertainty in AI, following methods are used:

Uncertain Knowledge and Reasoning in AI

- Dempster-Shafer Theory
- Probabilistic Reasoning
Hidden Markov Models (HMMs)
Belief Networks
- Hidden Markov Models (HMMs)
- Belief Networks
- Fuzzy Logic
- Neural Networks with dropout

Uncertain Knowledge and Reasoning in AI

- Hidden Markov Models (HMMs)

- Belief Networks

Learning in AI

Learning in Artificial Intelligence (AI) refers to the process by which a system improves its performance on a task over time through experience, data, or interaction with the environment.

Learning in AI

1. Supervised Learning: The model is trained on labeled dataset to learn the mapping from inputs to outputs.

Learning in AI

- Linear Regression
- Logistic Regression
- Support Vector Machines (SVM)
- Decision Trees
- Random Forests
- Neural Networks

Learning in AI

Semi-supervised learning uses both labeled and unlabeled data to improve learning accuracy.

Learning in AI

2. Unsupervised Learning: The model is trained on unlabeled dataset to discover patterns or structures.

Learning in AI

- K-Means Clustering
- Hierarchical Clustering
- Principal Component Analysis (PCA)
- Autoencoders

Learning in AI

3. Reinforcement Learning: The agent learns through interactions with an environment using feedbacks.

Learning in AI

- Q-Learning
- Deep Q-Networks (DQN)
- SARSA (State-Action-Reward-State-Action)
- Actor-Critic Methods

Learning in AI

4. Deep Learning: The concept focuses on using neural networks with many layers (hence "deep") to model and understand complex patterns and representations in large datasets.

Learning in AI

- Perceptron
- Artificial Neural Networks
- Activation Functions
- Recurrent Neural Network
- Convolutional Neural Network

Learning in AI

5. Probabilistic models in AI deal with uncertainty, making predictions, and modeling complex systems where uncertainty and variability play a crucial role. These models help in reasoning, decision-making, and learning from data.

Learning in AI

- Gaussian Mixture Models (GMMs)
- Naive Bayes Classifier

- Variational Inference
- Monte Carlo Methods
- Expectation-Maximization (EM) Algorithm

Communication, Perceiving, and Acting in AI and Robotics

Communication in AI and robotics facilitates interaction between machines and their environments, utilizing natural language processing. Perceiving involves machines using sensors and cameras to interpret their surroundings accurately. Acting in robotics includes making informed decisions and performing tasks based on processed data.

Communication, Perceiving, and Acting in AI and Robotics

- Natural Language Processing (NLP) Speech Recognition Natural Language Generation Chatbots Machine Translation
- Speech Recognition
- Natural Language Generation
- Chatbots
- Machine Translation
- Computer Vision Image Recognition Facial Recognition Optical Character Recognition
- Image Recognition
- Facial Recognition
- Optical Character Recognition
- Robotics

Communication, Perceiving, and Acting in AI and Robotics

- Speech Recognition
- Natural Language Generation
- Chatbots
- Machine Translation

Communication, Perceiving, and Acting in AI and Robotics

- Image Recognition
- Facial Recognition
- Optical Character Recognition

Generative AI

Generative AI focuses on creating new data instances that resemble real data, effectively learning the distribution of data to generate similar, but distinct, outputs.

Generative AI

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Diffusion Models
- Large Language Models

What is Artificial Intelligence?

Artificial Intelligence (AI) is a rapidly evolving field of computer science that focuses on creating intelligent machines capable of simulating human-like cognitive processes. At its core, AI seeks to enable machines to perceive their environment, learn from experience, reason, and make decisions autonomously. From virtual personal assistants and recommendation systems to autonomous vehicles and healthcare diagnostics, AI has become increasingly integrated into various aspects of our lives, revolutionizing industries and reshaping the way we interact with technology. As AI continues to advance, it holds the promise of solving complex problems, driving innovation, and transforming society in profound ways.

History of Artificial Intelligence

- 1940s-1950s: Early AI concepts emerged with neural networks and the Turing Test. The term "Artificial Intelligence" was coined at the 1956 Dartmouth Conference.
- 1960s-1970s: AI saw growth with developments like the first industrial robot and ELIZA, an early

chatbot. However, limitations in AI research led to the first AI winter in the 1970s.

- 1980s: AI research revived with expert systems and the popularization of backpropagation in neural networks. Despite this, the industry faced another downturn by the late 1980s.
- 1990s-2000s: Significant breakthroughs included IBM's Deep Blue defeating Garry Kasparov and the introduction of deep learning techniques. AI began integrating into consumer products, like the Roomba and early self-driving cars.
- 2010s: AI experienced a boom with advancements in deep learning, natural language processing (e.g., IBM's Watson), and AI-powered assistants. Google DeepMind's AlphaGo marked a significant milestone.
- 2020s: AI became widespread in everyday life, with innovations like GPT-3 and AlphaFold, alongside growing ethical concerns surrounding AI's impact.

History of Artificial Intelligence

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

History of Artificial Intelligence

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

History of Artificial Intelligence

- Artificial Intelligence
- AI-ML-DS
- Tutorials

Similar Reads

Similar Reads

- Company

- [About Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)

- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies

- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design

- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry

- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles

- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools

- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]

- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Prerequisites for Machine Learning

Getting Started with Machine Learning

Machine Learning Foundations

Packages For Machine Learning

Data Preprocessing

Creating Machine Learning Model

Machine Learning Deployment

Advance Topics in Machine Learning

Machine Learning Tutorial

-
-
-

Machine Learning Tutorial

Machine learning is a subset of Artificial Intelligence (AI) that enables computers to learn from data and make predictions without being explicitly programmed. If you're new to this field, this tutorial will provide a comprehensive understanding of machine learning, its types, algorithms, tools, and practical applications.

Module 1: Introduction to Machine Learning

Machine learning teaches computers to recognize patterns and make decisions automatically using data and algorithms.

Module 1: Introduction to Machine Learning

It can be broadly categorized into three types:

Module 1: Introduction to Machine Learning

- Supervised Learning: Trains models on labeled data to predict or classify new, unseen data.
- Unsupervised Learning: Finds patterns or groups in unlabeled data, like clustering or dimensionality reduction.
- Reinforcement Learning: Learns through trial and error to maximize rewards, ideal for decision-making tasks.

Module 1: Introduction to Machine Learning

In addition to these categories, there are also Semi-Supervised Learning and Self-Supervised Learning.

Module 1: Introduction to Machine Learning

- Semi-Supervised Learning uses a mix of labeled and unlabeled data, making it helpful when

labeling data is costly or time-consuming.

- Self-Supervised Learning creates its own labels from raw data, allowing it to learn patterns without needing labeled examples.

Machine Learning Pipeline

Machine learning is fundamentally built upon data, which serves as the foundation for training and testing models. Data consists of inputs (features) and outputs (labels). A model learns patterns during training and is tested on unseen data to evaluate its performance and generalization. In order to make predictions, there are essential steps through which data passes in order to produce a machine learning model that can make predictions.

Machine Learning Pipeline

- ML workflow
- Data Cleaning
- Feature Scaling
- Data Preprocessing in Python

Module 2: Supervised Learning

Supervised learning algorithms are generally categorized into two main types:

Module 2: Supervised Learning

- Classification- where the goal is to predict discrete labels or categories
- Regression- where the aim is to predict continuous numerical values.

Module 2: Supervised Learning

There are many algorithms used in supervised learning, each suited to different types of problems.

Some of the most commonly used supervised learning algorithms include:

1. Linear Regression

- Introduction to Linear Regression

- Gradient Descent in Linear Regression
- Linear regression (Python Implementation from scratch)
- Linear regression implementation using sklearn
- Rainfall prediction - Project
- Boston Housing Kaggle Challenge - Project
- Ridge Regression
- Lasso regression
- Elastic net Regression
- Implementation of Lasso, Ridge and Elastic Net

2. Logistic Regression

- Understanding Logistic Regression
- Cost function in Logistic Regression
- Logistic regression Implementation from scratch
- Heart Disease Prediction - Project
- Breast Cancer Wisconsin Diagnosis - Project

3. Decision Trees

- Decision Tree in Machine Learning
- Feature selection using Decision Tree
- Decision Tree - Regression (Implementation)
- Decision tree - Classification (Implementation)
- Types of Decision tree algorithms

4. Support Vector Machines (SVM)

- Understanding SVMs
- Support Vector Machines(SVMs) implementation
- SVM Hyperparameter Tuning - GridSearchCV

- Non-Linear SVM
- Implementing SVM on non-linear dataset

5. k-Nearest Neighbors (k-NN)

- Introduction to KNN
- Decision Boundaries in K-Nearest Neighbors (KNN)
- Implementation from scratch
- KNN classifier - Project

6. Naive Bayes

- Introduction to Naive Bayes
- Naive Bayes Scratch Implementation
- Gaussian Naive Bayes
- Implementation of Gaussian naive bayes
- Multinomial Naive Bayes
- Bernoulli Naive Bayes
- Complement Naive Bayes

Introduction to Ensemble Learning

Ensemble learning combines multiple simple models (called weak learners, like small decision trees) to create a stronger, smarter model. There are importantly two types of ensemble learning: Bagging that combines multiple models trained independently, and Boosting that builds models sequentially, each correcting the errors of the previous one.

Introduction to Ensemble Learning

For in-depth understanding :What is Ensemble Learning?-Two types of ensemble methods in ML

Introduction to Ensemble Learning

Advanced Supervised Learning Algorithms:

7. Random Forest (Bagging Algorithm)

- Introduction to Random forest
- Random Forest Classifier using Scikit-learn
- Random Forest Regression in Python
- Hyperparameter Tuning in Random Forest
- Credit Card Fraud Detection - Random Forest Classifier
- Voting Classifier

8. Boosting Algorithms

- Gradient Boosting in ML
- XGBoost (Extreme Gradient Boosting)
- LightGBM (Light Gradient Boosting Machine)
- CatBoost
- AdaBoost
- Gradient boosting regressor and XG implementation from scratch
- Calories Burnt Prediction - Project
- Tuning Hyperparameters in Gradient Boosting
- Box Office Revenue Prediction - Project
- Medical Insurance Price Prediction - Project
- Train a model using LightGBM
- Train a model using CatBoost
- E-commerce product recommendations using catboost - Project
- Implementing the AdaBoost Algorithm

8. Boosting Algorithms

Additionally, Stacking in machine learning is an ensemble learning technique that involves training multiple models (usually of different types) and combining their predictions using a meta-model, which learns the best way to combine the outputs of the individual models.

Module 3: Unsupervised learning

Unsupervised learning are again divided into three main categories based on their purpose: Clustering, Association Rule Mining, and Dimensionality Reduction. First we'll see algorithms for Clustering, then dimensionality reduction and at last, Association.

1. Clustering

Clustering algorithms group data points into clusters based on their similarities or differences. Clustering algorithms are divided into multiple types based on the methods they use to group data. These types include Centroid-based methods, Distribution-based methods, Connectivity-based methods, and Density-based methods. Let's understand each one of them:

1. Clustering

Centroid-based Methods: Represent clusters using central points, such as centroids or medoids.

1. Clustering

- K-Means clustering
- Elbow Method for optimal value of k in KMeans
- Clustering Text Documents using K-Means in Scikit Learn - Project
- Image Segmentation using K Means Clustering - Project
- KMeans Clustering and PCA on Wine Dataset - Project

1. Clustering

Modified versions of K-means algorithm:

1. Clustering

- K-Means++ clustering
- K-Mode clustering: Theory and implementation
- Fuzzy C-Means (FCM) Clustering
- Image Segmentation Using Fuzzy C-Means Clustering

1. Clustering

Distribution-based Methods:

1. Clustering

- Gaussian mixture models (GMMs)
- Implementation using GMM
- Expectation-Maximization Algorithm
- Dirichlet process mixture models (DPMMs)

1. Clustering

Connectivity based methods:

1. Clustering

- Hierarchical clustering
- Agglomerative Clustering
- Divisive clustering
- Implementing Agglomerative Clustering
- Affinity propagation

1. Clustering

Density Based methods:

1. Clustering

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- OPTICS (Ordering Points To Identify the Clustering Structure)

2. Dimensionality Reduction

Dimensionality reduction is used to simplify datasets by reducing the number of features while retaining the most important information.

2. Dimensionality Reduction

- Principal Component Analysis (PCA)
- Feature Importance in PCA
- Dimensionality Reduction with PCA : Implementation
- t-distributed Stochastic Neighbor Embedding (t-SNE)
- Non-negative Matrix Factorization (NMF)
- Handling Missing Values in NMF : Implementation
- Independent Component Analysis (ICA)
- Speech Separation Based On Fast ICA - Project
- FastICA on 2D Point Clouds in Scikit Learn - Project
- Isomap
- Locally Linear Embedding (LLE)
- Latent Semantic Analysis (LSA)
- Autoencoders

3. Association Rule

Find patterns (called association rules) between items in large datasets, typically in market basket analysis (e.g., finding that people who buy bread often buy butter). It identifies patterns based solely on the frequency of item occurrences and co-occurrences in the dataset.

3. Association Rule

- Apriori algorithm
- Implementing apriori algorithm
- FP-Growth (Frequent Pattern-Growth)
- ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal)

Module 4: Reinforcement Learning

Reinforcement learning are broadly categorized into Model-Based and Model-Free methods, these

approaches differ in how they interact with the environment.

1. Model-Based Methods

These methods use a model of the environment to predict outcomes and help the agent plan actions by simulating potential results.

1. Model-Based Methods

- Markov decision processes (MDPs)
- Bellman equation
- Value iteration algorithm
- Monte Carlo Tree Search

2. Model-Free Methods

These methods do not build or rely on an explicit model of the environment. Instead, the agent learns directly from experience by interacting with the environment and adjusting its actions based on feedback. Model-Free methods can be further divided into Value-Based and Policy-Based methods:

2. Model-Free Methods

Value-Based Methods: Focus on learning the value of different states or actions, where the agent estimates the expected return from each action and selects the one with the highest value.

2. Model-Free Methods

- Q-Learning
- SARSA
- Monte Carlo Methods

2. Model-Free Methods

Policy-based Methods: Directly learn a policy (a mapping from states to actions) without estimating values where the agent continuously adjusts its policy to maximize rewards.

2. Model-Free Methods

- Reinforce Algorithm
- Actor-Critic Algorithm
- Asynchronous Advantage Actor-Critic (A3C)

2. Model-Free Methods

Refer to 100+ Machine Learning Projects with Source Code [2024] for hands-on implementation projects

Module 5: Deployment of ML Models

The trained ML model must be integrated into an application or service to make its predictions accessible. Without integration, the model remains a theoretical artifact that cannot serve end-users. Let's learn how to deploy machine learning models into production. Everything you need to learn about Machine learning deployment

Module 5: Deployment of ML Models

End-users need a way to interact with the model, such as uploading data or viewing predictions. Using frameworks like Streamlit, Gradio, or custom-built web UIs.

Module 5: Deployment of ML Models

- Deploy ML web app (Streamlit) on Heroku
- Deploy ML Model using Streamlit Library
- Create UIs for prototyping Machine Learning model with Gradio

Module 5: Deployment of ML Models

Now, APIs allow other applications or systems to access the ML model's functionality programmatically, enabling automation and integration into larger workflows. Tools like FastAPI, Flask, or Django help create RESTful or gRPC endpoints that deliver predictions when called with appropriate input.

Module 5: Deployment of ML Models

- Deploy Machine Learning Model using Flask
- Deploying ML Models as API using FastAPI
- Django ? Machine Learning Placement Prediction Project
- Machine Learning Diabetes Prediction Project in Django

Module 6 : MLOps (Machine Learning Operations)

Learn how to operationalize Machine Learning models to ensure they are deployed, monitored, and maintained efficiently in real-world production systems.

Module 6 : MLOps (Machine Learning Operations)

- What is MLOps?
- Design Patterns in Machine Learning and MLOps
- MLOps Challenges
- Continuous Integration and Continuous Deployment (CI/CD) in MLOps
- End-to-End MLOps : Comprehensive Project
- MLOps Projects Ideas for beginners

Features of Machine Learning

- Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed.
- It is similar to data mining, as both deal with substantial amounts of data.
- For large organizations, branding is crucial, and targeting a relatable customer base becomes easier.
- Given a dataset, ML can detect various patterns in the data.
- Machines can learn from past data and automatically improve their performance.
- Machine learning is a data-driven technology. A large amount of data is generated by organizations daily, enabling them to identify notable relationships and make better decisions.

How ML is different from Deep learning ?

Machine learning develops programs that can access data and learn from it. Deep learning is the sub-domain of the machine learning. Deep learning supports automatic extraction of features from the raw data.

What are the next steps after learning machine learning?

Explore advanced topics like deep learning, natural language processing (NLP), and computer vision.

What are the next steps after learning machine learning?

- Deep Learning tutorial
- NLP tutorial
- Computer Vision Tutorial

How do I choose the right algorithm for a problem?

- Understand the type of problem (classification, regression, clustering, etc.).
- Consider the size and nature of your dataset.
- Experiment with multiple algorithms and evaluate them using metrics like accuracy, precision, recall, or RMSE.

What tools should I use for machine learning projects?

- Programming Languages: Python or R.
- Libraries: Scikit-learn, TensorFlow, PyTorch, Keras.
- Data Visualization: Matplotlib, Seaborn.
- Deployment: Flask, Docker, Kubernetes.

What tools should I use for machine learning projects?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

What tools should I use for machine learning projects?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

What tools should I use for machine learning projects?

- Machine Learning
- AI-ML-DS
- Tutorials
- Machine Learning

What tools should I use for machine learning projects?

- Machine Learning
- Machine Learning

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore

- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap

- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Deep Learning

- Introduction to Deep Learning
- Difference Between Artificial Intelligence vs Machine Learning vs Deep Learning

Basic Neural Network

- Difference between ANN and BNN
- Single Layer Perceptron in TensorFlow
- Multi-Layer Perceptron Learning in Tensorflow
- Deep Neural net with forward and back propagation from scratch - Python
- Understanding Multi-Layer Feed Forward Networks
- List of Deep Learning Layers

Activation Functions

- Activation Functions
- Types Of Activation Function in ANN
- Activation Functions in Pytorch
- Understanding Activation Functions in Depth

Artificial Neural Network

- Artificial Neural Networks and its Applications
- Gradient Descent Optimization in Tensorflow
- Choose Optimal Number of Epochs to Train a Neural Network in Keras

Classification

- Python | Classify Handwritten Digits with Tensorflow
- Train a Deep Learning Model With Pytorch

Regression

- Linear Regression using PyTorch
- Linear Regression Using Tensorflow

Hyperparameter tuning

- Hyperparameter tuning

Introduction to Convolution Neural Network

- Introduction to Convolution Neural Network
- Digital Image Processing Basics
- Difference between Image Processing and Computer Vision
- CNN | Introduction to Pooling Layer
- CIFAR-10 Image Classification in TensorFlow
- Implementation of a CNN based Image Classifier using PyTorch
- Convolutional Neural Network (CNN) Architectures
- Object Detection vs Object Recognition vs Image Segmentation
- YOLO v2 - Object Detection

Recurrent Neural Network

- Natural Language Processing (NLP) Tutorial
- Introduction to NLTK: Tokenization, Stemming, Lemmatization, POS Tagging
- Word Embeddings in NLP
- Introduction to Recurrent Neural Networks
- Recurrent Neural Networks Explanation
- Sentiment Analysis with an Recurrent Neural Networks (RNN)
- Short term Memory
- What is LSTM - Long Short Term Memory?
- Long Short Term Memory Networks Explanation
- LSTM - Derivation of Back propagation through time

- Text Generation using Recurrent Long Short Term Memory Network

Gated Recurrent Unit Networks

- Gated Recurrent Unit Networks
- ML | Text Generation using Gated Recurrent Unit Networks

Generative Learning

- Autoencoders -Machine Learning
- How Autoencoders works ?
- Variational AutoEncoders
- Contractive Autoencoder (CAE)
- ML | AutoEncoder with TensorFlow 2.0
- Implementing an Autoencoder in PyTorch

Generative adversarial networks

- Basics of Generative Adversarial Networks (GANs)
- Generative Adversarial Network (GAN)
- Use Cases of Generative Adversarial Networks
- Building a Generative Adversarial Network using Keras
- Cycle Generative Adversarial Network (CycleGAN)
- StyleGAN - Style Generative Adversarial Networks

Reinforcement Learning

- Understanding Reinforcement Learning in-depth
- Introduction to Thompson Sampling | Reinforcement Learning
- Markov Decision Process
- Bellman Equation
- Meta-Learning in Machine Learning

Q-Learning in Python

- Q-Learning
- ML | Reinforcement Learning Algorithm : Python Implementation using Q-learning

Deep Q Learning

- Deep Q-Learning
- Implementing Deep Q-Learning using Tensorflow
- AI Driven Snake Game using Deep Q Learning

Deep Learning Tutorial

-
-
-

Deep Learning Tutorial

Deep Learning tutorial covers the basics and more advanced topics, making it perfect for beginners and those with experience. Whether you're just starting or looking to expand your knowledge, this guide makes it easy to learn about the different technologies of Deep Learning.

Deep Learning Tutorial

- Deep Learning is a branch of Artificial Intelligence (AI) that enables machines to learn from large amounts of data.
- It uses neural networks with many layers to automatically find patterns and make predictions.
- It is very useful for tasks like image recognition, language translation, and speech processing.
- Deep learning models learn directly from data, without the need for manual feature extraction.
- Popular applications of Deep Learning include self-driving cars, chatbots, medical image analysis, and recommendation systems.

Introduction to Neural Networks

Neural Networks are fundamentals of deep learning inspired by human brain. It consists of layers of interconnected nodes, or "neurons," each designed to perform specific calculations. These nodes receive input data, process it through various mathematical functions, and pass the output to subsequent layers.

Introduction to Neural Networks

- Biological Neurons vs Artificial Neurons
- Single Layer Perceptron
- Multi-Layer Perceptron
- Artificial Neural Networks (ANNs)

Basic Components of Neural Networks

The basic components of neural network are:

Basic Components of Neural Networks

- Neurons
- Layers in Neural Networks
- Weights and Biases
- Forward Propagation
- Activation Functions
- Loss Functions
- Backpropagation
- Learning Rate

Optimization Algorithm in Deep Learning

Optimization algorithms in deep learning are used to minimize the loss function by adjusting the weights and biases of the model. The most common ones are:

Optimization Algorithm in Deep Learning

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Mini-batch Gradient Descent
- RMSprop (Root Mean Square Propagation)
- Adam (Adaptive Moment Estimation)

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep neural networks that are designed for processing grid-like data, such as images. They use convolutional layers to automatically detect patterns like edges, textures, and shapes in the data.

Convolutional Neural Networks (CNNs)

- Basics of Digital Image Processing
- Need for CNN
- Strides
- Padding
- Convolutional Layers
- Pooling Layers
- Fully Connected Layers
- Batch Normalization
- Backpropagation in CNNs

Convolutional Neural Networks (CNNs)

To learn about the implementation, you can explore the following articles:

Convolutional Neural Networks (CNNs)

- CNN based Image Classification using PyTorch
- CNN based Images Classification using TensorFlow

CNN Based Architectures

There are various architectures in CNNs that have been developed for specific kinds of problems, such as:

CNN Based Architectures

- LeNet-5
- AlexNet
- VGG-16 Network
- VGG-19 Network
- GoogLeNet/Inception
- ResNet (Residual Network)
- MobileNet

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of neural networks that are used for modeling sequence data such as time series or natural language.

Recurrent Neural Networks (RNNs)

- Vanishing Gradient and Exploding Gradient Problem
- How RNN Differs from Feedforward Neural Networks
- Backpropagation Through Time (BPTT)
- Types of Recurrent Neural Networks
- Bidirectional RNNs
- Long Short-Term Memory (LSTM)
- Bidirectional Long Short-Term Memory (Bi-LSTM)
- Gated Recurrent Units (GRU)

Generative Models in Deep Learning

Generative models generate new data that resembles the training data. The key types of generative

models include:

Generative Models in Deep Learning

- Generative Adversarial Networks (GANs)
- Autoencoders
- Restricted Boltzmann Machines (RBMs)

Variants of Generative Adversarial Networks (GANs)

GANs consists of two neural networks ? the generators and the discriminator that compete with each other in a game like framework. The variants of GANs include the following:

Variants of Generative Adversarial Networks (GANs)

- Deep Convolutional GAN (DCGAN)
- Conditional GAN (cGAN)
- Cycle-Consistent GAN (CycleGAN)
- Super-Resolution GAN (SRGAN)
- Wasserstein GAN (WGAN)
- StyleGAN

Types of Autoencoders

Autoencoders are neural networks used for unsupervised learning that learns to compress and reconstruct data. There are different types of autoencoders that serve different purpose such as noise reduction, generative modelling and feature learning.

Types of Autoencoders

- Sparse Autoencoder
- Denoising Autoencoder
- Undercomplete Autoencoder
- Contractive Autoencoder

- Convolutional Autoencoder
- Variational Autoencoder

Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning combines the representation learning power of deep learning with the decision-making ability of reinforcement learning. It enables agents to learn optimal behaviors in complex environments through trial and error, using high-dimensional sensory inputs.

Deep Reinforcement Learning (DRL)

- Reinforcement Learning
- Markov Decision Processes
- Function Approximation

Key Algorithms in Deep Reinforcement Learning

- Deep Q-Networks (DQN)
- REINFORCE
- Actor-Critic Methods
- Proximal Policy Optimization (PPO)

Key Algorithms in Deep Reinforcement Learning

Application of Deep Learning

- Image Recognition: Identifying objects, faces, and scenes in photos and videos.
- Natural Language Processing (NLP): Powering language translation, chatbots, and sentiment analysis.
- Speech Recognition: Converting spoken language into text for virtual assistants like Siri and Alexa.
- Medical Diagnostics: Detecting diseases from X-rays, MRIs, and other medical scans.
- Recommendation Systems: Personalizing suggestions for movies, music, and shopping.
- Autonomous Vehicles: Enabling self-driving cars to recognize objects and make driving decisions.

- Fraud Detection:Identifying unusual patterns in financial transactions and preventing fraud.
- Gaming:Enhancing AI in games and creating realistic environments in virtual reality.
- Predictive Analytics:Forecasting customer behavior, stock prices, and weather trends.
- Generative Models:Creating realistic images, deepfake videos, and AI-generated art.
- Robotics:Automating industrial tasks and powering intelligent drones.
- Customer Support:Enhancing chatbots for instant and intelligent customer interactions.

Which language is used for deep Learning?

Deep learning can be implemented using various programming languages, but some of the most commonly used ones are Python, C++, Java, and MATLAB.

What is the First Layer of Deep Learning?

The input layer is the first layer in any deep Learning Model.

How can I start learning deep learning?

You can easily start deep learning by following the given Steps:

How can I start learning deep learning?

- First, Learn machine learning basics.
- Start Learning Python.
- Choose a deep learning framework.
- Learn neural network basics.
- Practice with toy datasets.
- At Last, Work on real-world projects.

Is CNN deep learning?

Yes, Convolutional Neural Networks (CNNs) are a type of deep learning model commonly used in image recognition and computer vision tasks.

What is the difference between AI and deep learning?

Deep learning is a type of Artificial Intelligence and Machine learning that imitates the way humans gain certain types of knowledge.

What are the four pillars of Machine Learning?

The four pillars of deep learning are artificial neural networks, backpropagation, activation functions, and gradient descent.

Where can I practice Deep Learning interview questions?

You can prepare interview with our recommended Deep Learning Interview Question and answer

Where can I practice Deep Learning interview questions?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Where can I practice Deep Learning interview questions?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Where can I practice Deep Learning interview questions?

- AI-ML-DS
- Deep Learning
- AI-ML-DS With Python
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal

- [Privacy Policy](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)

- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS

- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science

- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies

- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor

- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Natural Language Processing (NLP) Tutorial

-
-
-

Natural Language Processing (NLP) Tutorial

Natural Language Processing (NLP) is the branch of Artificial Intelligence (AI) that gives the ability to machine understand and process human languages. Human languages can be in the form of text or audio format.

Applications of NLP

The applications of Natural Language Processing are as follows:

Applications of NLP

- Voice Assistants like Alexa, Siri, and Google Assistant use NLP for voice recognition and interaction.

- Tools like Grammarly, Microsoft Word, and Google Docs apply NLP for grammar checking and text analysis.
- Information extraction through Search engines such as Google and DuckDuckGo.
- Website bots and customer support chatbots leverage NLP for automated conversations and query handling.
- Google Translate and similar services use NLP for real-time translation between languages.
- Text summarization

Applications of NLP

This NLP tutorial is designed for both beginners and professionals. Whether you are a beginner or a data scientist, this guide will provide you with the knowledge and skills you need to take your understanding of NLP to the next level.

Phases of Natural Language Processing

Phases of Natural Language Processing

Phases of Natural Language Processing

There are two components of Natural Language Processing:

Phases of Natural Language Processing

- Natural Language Understanding
- Natural Language Generation

Libraries for Natural Language Processing

Some of natural language processing libraries include:

Libraries for Natural Language Processing

- NLTK (Natural Language Toolkit)
- spaCy

- Transformers (by Hugging Face)

- Gensim

Libraries for Natural Language Processing

To explore in detail, you can refer to this article: [NLP Libraries in Python](#)

Normalizing Textual Data in NLP

Text Normalization transforms text into a consistent format improves the quality and makes it easier to process in NLP tasks.

Normalizing Textual Data in NLP

Key steps in text normalization includes:

Normalizing Textual Data in NLP

1. Regular Expressions (RE) are sequences of characters that define search patterns.

Normalizing Textual Data in NLP

- How to write Regular Expressions?
- Properties of Regular Expressions
- RegEx in Python
- Email Extraction using RE

Normalizing Textual Data in NLP

2. Tokenization is a process of splitting text into smaller units called tokens.

Normalizing Textual Data in NLP

- How Tokenizing Text, Sentences, and Words Works
- Word Tokenization
- Rule-based Tokenization
- Subword Tokenization

- Dictionary-Based Tokenization
- Whitespace Tokenization
- WordPiece Tokenization

Normalizing Textual Data in NLP

3.Lemmatizationreduces words to their base or root form.

Normalizing Textual Data in NLP

4.Stemmingreduces words to their root by removing suffixes. Types of stemmers include:

Normalizing Textual Data in NLP

- Porter Stemmer
- Lancaster Stemmer
- Snowball Stemmer
- Lovis Stemmer
- Rule-based Stemming

Normalizing Textual Data in NLP

5.Stopword removalis a process to remove common words from the document.

Normalizing Textual Data in NLP

6.Parts of Speech (POS) Taggingassigns a part of speech to each word in sentence based on definition and context.

Text RepresentationorText Embedding Techniquesin NLP

Text representationconverts textual data into numerical vectors that are processed by the following methods:

Text RepresentationorText Embedding Techniquesin NLP

- One-Hot Encoding

- Bag of Words (BOW)
- N-Grams
- Term Frequency-Inverse Document Frequency (TF-IDF)
- N-Gram Language Modeling with NLTK

Text Representation or Text Embedding Techniques in NLP

Text Embedding Techniques refer to the methods and models used to create these vector representations, including traditional methods (like TFIDF and BOW) and more advanced approaches:

Text Representation or Text Embedding Techniques in NLP

1. Word Embedding

Text Representation or Text Embedding Techniques in NLP

- Word2Vec(SkipGram, Continuous Bag of Words ? CBOW)
- GloVe (Global Vectors for Word Representation)
- fastText

Text Representation or Text Embedding Techniques in NLP

2. Pre-Trained Embedding

Text Representation or Text Embedding Techniques in NLP

- ELMo (Embeddings from Language Models)
- BERT (Bidirectional Encoder Representations from Transformers)

Text Representation or Text Embedding Techniques in NLP

3. Document Embedding ? Doc2Vec

Deep Learning Techniques for NLP

Deep learning has revolutionized Natural Language Processing (NLP) by enabling models to

automatically learn complex patterns and representations from raw text. Below are some of the key deep learning techniques used in NLP:

Deep Learning Techniques for NLP

- Artificial Neural Networks (ANNs)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- Seq2Seq Models
- Transformer Models

Pre-Trained Language Models

Pre-trained models understand language patterns, context and semantics. The provided models are trained on massive corpora and can be fine tuned for specific tasks.

Pre-Trained Language Models

- GPT (Generative Pre-trained Transformer)
- Transformers XL
- T5 (Text-to-Text Transfer Transformer)
- RoBERTa

Pre-Trained Language Models

To learn how to fine tune a model, refer to this article: [Transfer Learning with Fine-tuning](#)

Natural Language Processing Tasks

1. Text Classification

Natural Language Processing Tasks

- Dataset for Text Classification
- Text Classification using Naive Bayes

- Text Classification using Logistic Regression
- Text Classification using RNNs
- Text Classification using CNNs

Natural Language Processing Tasks

2. Information Extraction

Natural Language Processing Tasks

- Information Extraction
- Named Entity Recognition (NER) using SpaCy
- Named Entity Recognition (NER) using NLTK
- Relationship Extraction

Natural Language Processing Tasks

3. Sentiment Analysis

Natural Language Processing Tasks

- What is Sentiment Analysis?
- Sentiment Analysis using VADER
- Sentiment Analysis using Recurrent Neural Networks (RNN)

Natural Language Processing Tasks

4. Machine Translation

Natural Language Processing Tasks

- Statistical Machine Translation of Language
- Machine Translation with Transformer

Natural Language Processing Tasks

5. Text Summarization

Natural Language Processing Tasks

- What is Text Summarization?
- Text Summarizations using Hugging Face Model
- Text Summarization using Sumy

Natural Language Processing Tasks

6. Text Generation

Natural Language Processing Tasks

- Text Generation using Fnet
- Text Generation using Recurrent Long Short Term Memory Network
- Text2Text Generations using HuggingFace Model

History of NLP

Natural Language Processing (NLP) emerged in 1950 when Alan Turing published his groundbreaking paper titled *Computing Machinery and Intelligence*. Turing's work laid the foundation for NLP, which is a subset of Artificial Intelligence (AI) focused on enabling machines to automatically interpret and generate human language. Over time, NLP technology has evolved, giving rise to different approaches for solving complex language-related tasks.

1. Heuristic-Based NLP

The Heuristic-based approach to NLP was one of the earliest methods used in natural language processing. It relies on predefined rules and domain-specific knowledge. These rules are typically derived from expert insights. A classic example of this approach is Regular Expressions (Regex), which are used for pattern matching and text manipulation tasks.

2. Statistical and Machine Learning-Based NLP

As NLP advanced, Statistical NLP emerged, incorporating machine learning algorithms to model language patterns. This approach applies statistical rules and learns from data to tackle various

language processing tasks. Popular machine learning algorithms in this category include:

2. Statistical and Machine Learning-Based NLP

- Naive Bayes
- Support Vector Machines (SVM)
- Hidden Markov Models (HMM)

3. Neural Network-Based NLP (Deep Learning)

The most recent advancement in NLP is the adoption of Deep Learning techniques. Neural networks, particularly Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), and Transformers, have revolutionized NLP tasks by providing superior accuracy. These models require large amounts of data and considerable computational power for training.

What is the most difficult part of natural language processing?

Ambiguity is the main challenge of natural language processing because in natural language, words are unique, but they have different meanings depending upon the context which causes ambiguity on lexical, syntactic, and semantic levels.

What are the 4 pillars of NLP?

The four main pillars of NLP are 1.) Outcomes, 2.) Sensory acuity, 3.) behavioural flexibility, and 4.) report.

What language is best for natural language processing?

Python is considered the best programming language for NLP because of their numerous libraries, simple syntax, and ability to easily integrate with other programming languages.

What is the life cycle of NLP?

There are four stages included in the life cycle of NLP : development, validation, deployment, and monitoring of the models.

What is the life cycle of NLP?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

What is the life cycle of NLP?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

What is the life cycle of NLP?

- AI-ML-DS
- Data Science
- Machine Learning
- NLP
- Natural-language-processing
- python

What is the life cycle of NLP?

- Machine Learning
- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media

- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS

- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git

- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter

- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course

- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification

- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is a Large Language Model (LLM)

-
-
-

What is a Large Language Model (LLM)

Large Language Models (LLMs) represent a breakthrough in artificial intelligence, employing neural network techniques with extensive parameters for advanced language processing.

What is a Large Language Model (LLM)

This article explores the evolution, architecture, applications, and challenges of LLMs, focusing on their impact in the field of Natural Language Processing (NLP).

What are Large Language Models(LLMs)?

A large language model is a type of artificial intelligence algorithm that applies neural network techniques with lots of parameters to process and understand human languages or text using self-supervised learning techniques. Tasks like text generation, machine translation, summary writing, image generation from texts, machine coding, chat-bots, or Conversational AI are applications of the Large Language Model.

What are Large Language Models(LLMs)?

Examples of such LLM models are Chat GPT by open AI, BERT (Bidirectional Encoder Representations from Transformers) by Google, etc.

What are Large Language Models(LLMs)?

There are many techniques that were tried to perform natural language-related tasks but the LLM is purely based on the deep learning methodologies. LLM (Large language model) models are highly efficient in capturing the complex entity relationships in the text at hand and can generate the text using the semantic and syntactic of that particular language in which we wish to do so.

What are Large Language Models(LLMs)?

If we talk about the size of the advancements in the GPT (Generative Pre-trained Transformer) model only then:

What are Large Language Models(LLMs)?

- GPT-1 which was released in 2018 contains 117 million parameters having 985 million words.
- GPT-2 which was released in 2019 contains 1.5 billion parameters.
- GPT-3 which was released in 2020 contains 175 billion parameters. Chat GPT is also based on this model as well.
- GPT-4 model is released in the early 2023 and it is likely to contain trillions of parameters.
- GPT-4 Turbo was introduced in late 2023, optimized for speed and cost-efficiency, but its parameter count remains unspecified.

How do Large Language Models work?

Large Language Models (LLMs) operate on the principles of deep learning, leveraging neural network architectures to process and understand human languages.

How do Large Language Models work?

These models, are trained on vast datasets using self-supervised learning techniques. The core of

their functionality lies in the intricate patterns and relationships they learn from diverse language data during training. LLMs consist of multiple layers, including feedforward layers, embedding layers, and attention layers. They employ attention mechanisms, like self-attention, to weigh the importance of different tokens in a sequence, allowing the model to capture dependencies and relationships.

Architecture of LLM

Large Language Model's (LLM) architecture is determined by a number of factors, like the objective of the specific model design, the available computational resources, and the kind of language processing tasks that are to be carried out by the LLM. The general architecture of LLM consists of many layers such as the feed forward layers, embedding layers, attention layers. A text which is embedded inside is collaborated together to generate predictions.

Architecture of LLM

Important components to influence Large Language Model architecture:

Architecture of LLM

- Model Size and Parameter Count
- input representations
- Self-Attention Mechanisms
- Training Objectives
- Computational Efficiency
- Decoding and Output Generation

Transformer-Based LLM Model Architectures

Transformer-based models, which have revolutionized natural language processing tasks, typically follow a general architecture that includes the following components:

Transformer-Based LLM Model Architectures

Transformer-Based LLM Model Architectures

- **Input Embeddings:**The input text is tokenized into smaller units, such as words or sub-words, and each token is embedded into a continuous vector representation. This embedding step captures the semantic and syntactic information of the input.
- **Positional Encoding:**Positional encoding is added to the input embeddings to provide information about the positions of the tokens because transformers do not naturally encode the order of the tokens. This enables the model to process the tokens while taking their sequential order into account.
- **Encoder:**Based on a neural network technique, the encoder analyses the input text and creates a number of hidden states that protect the context and meaning of text data. Multiple encoder layers make up the core of the transformer architecture. Self-attention mechanism and feed-forward neural network are the two fundamental sub-components of each encoder layer.
Self-Attention Mechanism:Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.
Feed-Forward Neural Network:After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.
- **Self-Attention Mechanism:**Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.
- **Feed-Forward Neural Network:**After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.
- **Decoder Layers:**In some transformer-based models, a decoder component is included in addition to the encoder. The decoder layers enable autoregressive generation, where the model can generate sequential outputs by attending to the previously generated tokens.

- **Multi-Head Attention:** Transformers often employ multi-head attention, where self-attention is performed simultaneously with different learned attention weights. This allows the model to capture different types of relationships and attend to various parts of the input sequence simultaneously.
- **Layer Normalization:** Layer normalization is applied after each sub-component or layer in the transformer architecture. It helps stabilize the learning process and improves the model's ability to generalize across different inputs.
- **Output Layers:** The output layers of the transformer model can vary depending on the specific task. For example, in language modeling, a linear projection followed by SoftMax activation is commonly used to generate the probability distribution over the next token.

Transformer-Based LLM Model Architectures

- **Self-Attention Mechanism:** Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.
- **Feed-Forward Neural Network:** After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.

Transformer-Based LLM Model Architectures

It's important to keep in mind that the actual architecture of transformer-based models can change and be enhanced based on particular research and model creations. To fulfill different tasks and objectives, several models like GPT, BERT, and T5 may integrate more components or modifications.

Popular Large Language Models

Now let's look at some of the famous LLMs which has been developed and are up for inference.

Popular Large Language Models

- **GPT-3:** GPT 3 is developed by OpenAI, stands for Generative Pre-trained Transformer 3. This

model powers ChatGPT and is widely recognized for its ability to generate human-like text across a variety of applications.

- BERT: It is created by Google, is commonly used for natural language processing tasks and generating text embeddings, which can also be utilized for training other models.

- RoBERTa: RoBERTa is an advanced version of BERT, stands for Robustly Optimized BERT Pretraining Approach. Developed by Facebook AI Research, it enhances the performance of the transformer architecture.

- BLOOM: It is the first multilingual LLM, designed collaboratively by multiple organizations and researchers. It follows an architecture similar to GPT-3, enabling diverse language-based tasks.

Popular Large Language Models

For implementation details, these models are available on open-source platforms like Hugging Face and OpenAI for Python-based applications.

Large Language Models Use Cases

- Code Generation: LLMs can generate accurate code based on user instructions for specific tasks.
- Debugging and Documentation: They assist in identifying code errors, suggesting fixes, and even automating project documentation.
- Question Answering: Users can ask both casual and complex questions, receiving detailed, context-aware responses.
- Language Translation and Correction: LLMs can translate text between over 50 languages and correct grammatical errors.
- Prompt-Based Versatility: By crafting creative prompts, users can unlock endless possibilities, as LLMs excel in one-shot and zero-shot learning scenarios.

Large Language Models Use Cases

Use cases of LLM are not limited to the above-mentioned one has to be just creative enough to write better prompts and you can make these models do a variety of tasks as they are trained to

perform tasks on one-shot learning and zero-shot learning methodologies as well. Due to this only Prompt Engineering is a totally new and hot topic in academics for people who are looking forward to using ChatGPT-type models extensively.

Applications of Large Language Models

LLMs, such as GPT-3, have a wide range of applications across various domains. Few of them are:

Applications of Large Language Models

- Natural Language Understanding (NLU): Large language models power advanced chatbots capable of engaging in natural conversations. They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and information retrieval.
- Large language models power advanced chatbots capable of engaging in natural conversations.
- They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and information retrieval.
- Content Generation: Creating human-like text for various purposes, including content creation, creative writing, and storytelling. Writing code snippets based on natural language descriptions or commands.
- Creating human-like text for various purposes, including content creation, creative writing, and storytelling.
- Writing code snippets based on natural language descriptions or commands.
- Language Translation: Large language models can aid in translating text between different languages with improved accuracy and fluency.
- Text Summarization: Generating concise summaries of longer texts or articles.
- Sentiment Analysis: Analyzing and understanding sentiments expressed in social media posts, reviews, and comments.

Applications of Large Language Models

- Large language models power advanced chatbots capable of engaging in natural conversations.

- They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and information retrieval.

Applications of Large Language Models

- Creating human-like text for various purposes, including content creation, creative writing, and storytelling.
- Writing code snippets based on natural language descriptions or commands.

Difference Between NLP and LLM

NLP is Natural Language Processing, a field of artificial intelligence (AI). It consists of the development of the algorithms. NLP is a broader field than LLM, which consists of algorithms and techniques. NLP rules two approaches i.e. Machine learning and the analyze language data. Applications of NLP are-

Difference Between NLP and LLM

- Automative routine task
- Improve search
- Search engine optimization
- Analyzing and organizing large documents
- Social Media Analytics.

Difference Between NLP and LLM

while on the other hand, LLM is a Large Language Model, and is more specific to human- like text, providing content generation, and personalized recommendations.

What are the Advantages of Large Language Models?

Large Language Models (LLMs) come with several advantages that contribute to their widespread adoption and success in various applications:

What are the Advantages of Large Language Models?

- LLMs can perform zero-shot learning, meaning they can generalize to tasks for which they were not explicitly trained. This capability allows for adaptability to new applications and scenarios without additional training.
- LLMs efficiently handle vast amounts of data, making them suitable for tasks that require a deep understanding of extensive text corpora, such as language translation and document summarization.
- LLMs can be fine-tuned on specific datasets or domains, allowing for continuous learning and adaptation to specific use cases or industries.
- LLMs enable the automation of various language-related tasks, from code generation to content creation, freeing up human resources for more strategic and complex aspects of a project.

Challenges in Training of Large Language Models

- High Costs: Training LLMs requires significant financial investment, with millions of dollars needed for large-scale computational power.
- Time-Intensive: Training takes months, often involving human intervention for fine-tuning to achieve optimal performance.
- Data Challenges: Obtaining large text datasets is difficult, and concerns about the legality of data scraping for commercial purposes have arisen.
- Environmental Impact: Training a single LLM from scratch can produce carbon emissions equivalent to the lifetime emissions of five cars, raising serious environmental concerns.

Conclusion

Due to the challenges faced in training LLM transfer learning is promoted heavily to get rid of all of the challenges discussed above. LLM has the capability to bring revolution in the AI-powered application but the advancements in this field seem a bit difficult because just increasing the size of the model may increase its performance but after a particular time a saturation in the performance will come and the challenges to handle these models will be bigger than the performance boost achieved by further increasing the size of the models.

What is a large language model?

A large language model is a powerful artificial intelligence system trained on vast amounts of text data.

What is a LLM in AI?

In AI, LLM refers to Large Language Models, such as GPT-3, designed for natural language understanding and generation.

What are the best Large Language Models?

Open AI, ChatGPT, GPT-3, GooseAI, Claude, Cohere, GPT-4.

How does LLM model work?

LLMs work by training on diverse language data, learning patterns, and relationships, enabling them to understand and generate human-like text.

What is an example of an LLM model?

GPT-3 (Generative Pre-trained Transformer 3) is an example of a state-of-the-art large language model in AI.

What are large language models for education?

Large Language Models are widely being used for educational purposes:

What are large language models for education?

- Provides learning goals
- Gives a critical summary of any topic to the students
- Educate students on any topic they want to learn.

What are large language models for education?

What are large language models for education?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90

Challenge today.

What are large language models for education?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

What are large language models for education?

A

What are large language models for education?

- AI-ML-DS
- Data Science
- Machine Learning
- NLP
- ChatGPT
- data-science

What are large language models for education?

- Machine Learning

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us

- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS

- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS

- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce

- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams

- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator

- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course

- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is a Large Language Model (LLM)

-
-
-

What is a Large Language Model (LLM)

Large Language Models (LLMs) represent a breakthrough in artificial intelligence, employing neural network techniques with extensive parameters for advanced language processing.

What is a Large Language Model (LLM)

This article explores the evolution, architecture, applications, and challenges of LLMs, focusing on their impact in the field of Natural Language Processing (NLP).

What are Large Language Models(LLMs)?

A large language model is a type of artificial intelligence algorithm that applies neural network techniques with lots of parameters to process and understand human languages or text using self-supervised learning techniques. Tasks like text generation, machine translation, summary writing, image generation from texts, machine coding, chat-bots, or Conversational AI are applications of the Large Language Model.

What are Large Language Models(LLMs)?

Examples of such LLM models are Chat GPT by open AI, BERT (Bidirectional Encoder Representations from Transformers) by Google, etc.

What are Large Language Models(LLMs)?

There are many techniques that were tried to perform natural language-related tasks but the LLM is purely based on the deep learning methodologies. LLM (Large language model) models are highly efficient in capturing the complex entity relationships in the text at hand and can generate the text using the semantic and syntactic of that particular language in which we wish to do so.

What are Large Language Models(LLMs)?

If we talk about the size of the advancements in the GPT (Generative Pre-trained Transformer) model only then:

What are Large Language Models(LLMs)?

- GPT-1 which was released in 2018 contains 117 million parameters having 985 million words.
- GPT-2 which was released in 2019 contains 1.5 billion parameters.
- GPT-3 which was released in 2020 contains 175 billion parameters. Chat GPT is also based on this model as well.
- GPT-4 model is released in the early 2023 and it is likely to contain trillions of parameters.
- GPT-4 Turbo was introduced in late 2023, optimized for speed and cost-efficiency, but its parameter count remains unspecified.

How do Large Language Models work?

Large Language Models (LLMs) operate on the principles of deep learning, leveraging neural network architectures to process and understand human languages.

How do Large Language Models work?

These models, are trained on vast datasets using self-supervised learning techniques. The core of their functionality lies in the intricate patterns and relationships they learn from diverse language

data during training. LLMs consist of multiple layers, including feedforward layers, embedding layers, and attention layers. They employ attention mechanisms, like self-attention, to weigh the importance of different tokens in a sequence, allowing the model to capture dependencies and relationships.

Architecture of LLM

Large Language Model's (LLM) architecture is determined by a number of factors, like the objective of the specific model design, the available computational resources, and the kind of language processing tasks that are to be carried out by the LLM. The general architecture of LLM consists of many layers such as the feed forward layers, embedding layers, attention layers. A text which is embedded inside is collaborated together to generate predictions.

Architecture of LLM

Important components to influence Large Language Model architecture:

Architecture of LLM

- Model Size and Parameter Count
- input representations
- Self-Attention Mechanisms
- Training Objectives
- Computational Efficiency
- Decoding and Output Generation

Transformer-Based LLM Model Architectures

Transformer-based models, which have revolutionized natural language processing tasks, typically follow a general architecture that includes the following components:

Transformer-Based LLM Model Architectures

Transformer-Based LLM Model Architectures

- **Input Embeddings:**The input text is tokenized into smaller units, such as words or sub-words, and each token is embedded into a continuous vector representation. This embedding step captures the semantic and syntactic information of the input.
- **Positional Encoding:**Positional encoding is added to the input embeddings to provide information about the positions of the tokens because transformers do not naturally encode the order of the tokens. This enables the model to process the tokens while taking their sequential order into account.
- **Encoder:**Based on a neural network technique, the encoder analyses the input text and creates a number of hidden states that protect the context and meaning of text data. Multiple encoder layers make up the core of the transformer architecture. Self-attention mechanism and feed-forward neural network are the two fundamental sub-components of each encoder layer.
- Self-Attention Mechanism:**Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.
- Feed-Forward Neural Network:**After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.
- **Self-Attention Mechanism:**Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.
- **Feed-Forward Neural Network:**After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.
- **Decoder Layers:**In some transformer-based models, a decoder component is included in addition to the encoder. The decoder layers enable autoregressive generation, where the model can generate sequential outputs by attending to the previously generated tokens.
- **Multi-Head Attention:**Transformers often employ multi-head attention, where self-attention is

performed simultaneously with different learned attention weights. This allows the model to capture different types of relationships and attend to various parts of the input sequence simultaneously.

- Layer Normalization: Layer normalization is applied after each sub-component or layer in the transformer architecture. It helps stabilize the learning process and improves the model's ability to generalize across different inputs.

- Output Layers: The output layers of the transformer model can vary depending on the specific task. For example, in language modeling, a linear projection followed by SoftMax activation is commonly used to generate the probability distribution over the next token.

Transformer-Based LLM Model Architectures

- Self-Attention Mechanism: Self-attention enables the model to weigh the importance of different tokens in the input sequence by computing attention scores. It allows the model to consider the dependencies and relationships between different tokens in a context-aware manner.

- Feed-Forward Neural Network: After the self-attention step, a feed-forward neural network is applied to each token independently. This network includes fully connected layers with non-linear activation functions, allowing the model to capture complex interactions between tokens.

Transformer-Based LLM Model Architectures

It's important to keep in mind that the actual architecture of transformer-based models can change and be enhanced based on particular research and model creations. To fulfill different tasks and objectives, several models like GPT, BERT, and T5 may integrate more components or modifications.

Popular Large Language Models

Now let's look at some of the famous LLMs which has been developed and are up for inference.

Popular Large Language Models

- GPT-3: GPT 3 is developed by OpenAI, stands for Generative Pre-trained Transformer 3. This model powers ChatGPT and is widely recognized for its ability to generate human-like text across a

variety of applications.

- BERT: It is created by Google, is commonly used for natural language processing tasks and generating text embeddings, which can also be utilized for training other models.
- RoBERTa: RoBERTa is an advanced version of BERT, stands for Robustly Optimized BERT Pretraining Approach. Developed by Facebook AI Research, it enhances the performance of the transformer architecture.
- BLOOM: It is the first multilingual LLM, designed collaboratively by multiple organizations and researchers. It follows an architecture similar to GPT-3, enabling diverse language-based tasks.

Popular Large Language Models

For implementation details, these models are available on open-source platforms like Hugging Face and OpenAI for Python-based applications.

Large Language Models Use Cases

- Code Generation: LLMs can generate accurate code based on user instructions for specific tasks.
- Debugging and Documentation: They assist in identifying code errors, suggesting fixes, and even automating project documentation.
- Question Answering: Users can ask both casual and complex questions, receiving detailed, context-aware responses.
- Language Translation and Correction: LLMs can translate text between over 50 languages and correct grammatical errors.
- Prompt-Based Versatility: By crafting creative prompts, users can unlock endless possibilities, as LLMs excel in one-shot and zero-shot learning scenarios.

Large Language Models Use Cases

Use cases of LLM are not limited to the above-mentioned one has to be just creative enough to write better prompts and you can make these models do a variety of tasks as they are trained to perform tasks on one-shot learning and zero-shot learning methodologies as well. Due to this only

Prompt Engineering is a totally new and hot topic in academics for people who are looking forward to using ChatGPT-type models extensively.

Applications of Large Language Models

LLMs, such as GPT-3, have a wide range of applications across various domains. Few of them are:

Applications of Large Language Models

- Natural Language Understanding (NLU): Large language models power advanced chatbots capable of engaging in natural conversations. They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and information retrieval.
- Large language models power advanced chatbots capable of engaging in natural conversations.
- They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and information retrieval.
- Content Generation: Creating human-like text for various purposes, including content creation, creative writing, and storytelling. Writing code snippets based on natural language descriptions or commands.
- Creating human-like text for various purposes, including content creation, creative writing, and storytelling.
- Writing code snippets based on natural language descriptions or commands.
- Language Translation: Large language models can aid in translating text between different languages with improved accuracy and fluency.
- Text Summarization: Generating concise summaries of longer texts or articles.
- Sentiment Analysis: Analyzing and understanding sentiments expressed in social media posts, reviews, and comments.

Applications of Large Language Models

- Large language models power advanced chatbots capable of engaging in natural conversations.
- They can be used to create intelligent virtual assistants for tasks like scheduling, reminders, and

information retrieval.

Applications of Large Language Models

- Creating human-like text for various purposes, including content creation, creative writing, and storytelling.
- Writing code snippets based on natural language descriptions or commands.

Difference Between NLP and LLM

NLP is Natural Language Processing, a field of artificial intelligence (AI). It consists of the development of the algorithms. NLP is a broader field than LLM, which consists of algorithms and techniques. NLP rules two approaches i.e. Machine learning and the analyze language data. Applications of NLP are-

Difference Between NLP and LLM

- Automotive routine task
- Improve search
- Search engine optimization
- Analyzing and organizing large documents
- Social Media Analytics.

Difference Between NLP and LLM

while on the other hand, LLM is a Large Language Model, and is more specific to human- like text, providing content generation, and personalized recommendations.

What are the Advantages of Large Language Models?

Large Language Models (LLMs) come with several advantages that contribute to their widespread adoption and success in various applications:

What are the Advantages of Large Language Models?

- LLMs can perform zero-shot learning, meaning they can generalize to tasks for which they were not

explicitly trained. This capability allows for adaptability to new applications and scenarios without additional training.

- LLMs efficiently handle vast amounts of data, making them suitable for tasks that require a deep understanding of extensive text corpora, such as language translation and document summarization.
- LLMs can be fine-tuned on specific datasets or domains, allowing for continuous learning and adaptation to specific use cases or industries.
- LLMs enable the automation of various language-related tasks, from code generation to content creation, freeing up human resources for more strategic and complex aspects of a project.

Challenges in Training of Large Language Models

- High Costs: Training LLMs requires significant financial investment, with millions of dollars needed for large-scale computational power.
- Time-Intensive: Training takes months, often involving human intervention for fine-tuning to achieve optimal performance.
- Data Challenges: Obtaining large text datasets is difficult, and concerns about the legality of data scraping for commercial purposes have arisen.
- Environmental Impact: Training a single LLM from scratch can produce carbon emissions equivalent to the lifetime emissions of five cars, raising serious environmental concerns.

Conclusion

Due to the challenges faced in training LLM transfer learning is promoted heavily to get rid of all of the challenges discussed above. LLM has the capability to bring revolution in the AI-powered application but the advancements in this field seem a bit difficult because just increasing the size of the model may increase its performance but after a particular time a saturation in the performance will come and the challenges to handle these models will be bigger than the performance boost achieved by further increasing the size of the models.

What is a large language model?

A large language model is a powerful artificial intelligence system trained on vast amounts of text data.

What is a LLM in AI?

In AI, LLM refers to Large Language Models, such as GPT-3, designed for natural language understanding and generation.

What are the best Large Language Models?

Open AI, ChatGPT, GPT-3, GooseAI, Claude, Cohere, GPT-4.

How does LLM model work?

LLMs work by training on diverse language data, learning patterns, and relationships, enabling them to understand and generate human-like text.

What is an example of an LLM model?

GPT-3 (Generative Pre-trained Transformer 3) is an example of a state-of-the-art large language model in AI.

What are large language models for education?

Large Language Models are widely being used for educational purposes:

What are large language models for education?

- Provides learning goals
- Gives a critical summary of any topic to the students
- Educate students on any topic they want to learn.

What are large language models for education?

What are large language models for education?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90

Challenge today.

What are large language models for education?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

What are large language models for education?

A

What are large language models for education?

- AI-ML-DS
- Data Science
- Machine Learning
- NLP
- ChatGPT
- data-science

What are large language models for education?

- Machine Learning

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us

- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS

- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS

- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce

- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams

- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator

- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course

- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to React-Redux

-
-
-

Introduction to React-Redux

React is a popular JavaScript library for building user interfaces, especially for single-page applications. It helps developers create fast, scalable, and flexible front-end applications. However, as your React app grows, managing the state can become challenging. This is where Redux comes into play.

Introduction to React-Redux

In this article, we will see the introduction to React-Redux, covering what it is, how it works, and why you should use it in your next project. We'll also walk through a simple React-Redux tutorial to give you hands-on experience with this powerful state management solution.

What Is React-Redux?

React-Redux is the official Redux binding for React. It allows React components to read data from a Redux store and dispatch actions to update the store. Essentially, React-Redux provides a way to manage and share the application's state across multiple components in a React application.

What Is React-Redux?

Redux on its own is not tied to React; it can be used with any JavaScript framework or library. However, React-Redux makes it easier to connect Redux with React by providing a set of utilities that handle common tasks such as accessing the store and dispatching actions.

Key Features of React-Redux:

- Centralized State Management: Redux stores the entire state of the application in a single store, making state management easier.
- Predictable State Updates: Redux updates the state in a predictable way through actions and reducers, allowing for more consistent application behaviour.
- Component Separation: With React-Redux, your UI components and state management are decoupled, making your code cleaner and easier to maintain.

Why Use React-Redux?

Managing the state in React applications with prop drilling or React's built-in state can become difficult as the app scales. React-Redux solves this issue by centralizing the state, allowing components to subscribe to only the data they need. This helps in reducing prop drilling and makes the code more maintainable.

Why Use React-Redux?

Here are a few benefits of using React-Redux:

Why Use React-Redux?

- Better State Management: Redux centralizes and manages application-wide state in a predictable manner, making it easier to track and update.
- Debugging and Testing: Redux's predictable state updates help in debugging, as state changes are consistent and can be traced through dispatched actions.
- Code Maintainability: With actions, reducers, and the Redux store, React-Redux organizes your code into distinct, manageable pieces, improving maintainability over time.

Step 1: Setting Up the Project

First, create a new React app using create-react-app:

Step 1: Setting Up the Project

Next, install redux and react-redux:

Folder Structure

Folder Structure

Step 2: Create the Redux Slice (counterSlice.js)

In the src/redux/ folder, create a file called counterSlice.js. This will define the actions and reducer for the counter.

Step 3: Create the Redux Store (store.js)

Now, create the Redux store in the src/redux/store.js file:

Step 4: Create the Counter Component (Counter.js)

Now, create a simple Counter component in the src/components/Counter.js file. This component will access the state and dispatch actions to increment or decrement the count.

Step 5: Connect Redux to React

Next, wrap your App component in a Redux Provider so that the store is available throughout the app.

Step 5: Connect Redux to React

Step 5: Connect Redux to React

To start the application run the following command.

Output

Introduction to React-Redux

Output

Become a Full-Stack Developer and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Output

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Output

L

Output

- ReactJS
- Web Technologies
- Redux

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners

- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025

- GATE DA Course 2025

Similar Reads

Ollama Explained: Transforming AI Accessibility and Language Processing

-
-
-

Ollama Explained: Transforming AI Accessibility and Language Processing

In the rapidly evolving landscape of artificial intelligence (AI), accessibility and innovation are paramount. Among the myriad platforms and tools emerging in this space, one name stands out: Ollama. But what exactly is Ollama, and why is it garnering attention in the AI community? This article delves into the intricacies of Ollama, its methodologies, its potential impact on AI applications, and what this could mean for the future of human-machine interaction.

Ollama Explained: Transforming AI Accessibility and Language Processing

Table of Content

Ollama Explained: Transforming AI Accessibility and Language Processing

- Understanding Ollama
- The Genesis of Ollama:
- Understanding the Ollama Framework:
- Stepwise Guide to start Ollama
- Applications of Ollama
- Ethical Considerations and Responsible AI
- Conclusion

Ollama Explained: Transforming AI Accessibility and Language Processing

Understanding Ollama

Ollama stands for (Omni-Layer Learning Language Acquisition Model), a novel approach to machine learning that promises to redefine how we perceive language acquisition and natural language processing. At its core, Ollama is a groundbreaking platform that democratizes access to large language models (LLMs) by enabling users to run them locally on their machines. Developed with a vision to empower individuals and organizations, Ollama provides a user-friendly interface and seamless integration capabilities, making it easier than ever to leverage the power of LLMs for various applications and use cases.

The Genesis of Ollama:

Traditional machine learning models have been successful in various linguistic tasks but often require extensive data labelling and preprocessing to function effectively. Ollama emerges as a paradigm shift, utilizing unsupervised learning techniques combined with deep neural networks that enable it to learn language structures without explicit grammatical rules or annotations.

Understanding the Ollama Framework:

OLLAMA's architecture comprises multiple layers where each successive layer learns different linguistic patterns and abstract representations of speech. This multi-layered approach allows OLLAMA to progress from understanding basic sounds to grasping complex sentence structures, all without direct human intervention for labelling or structuring the input data.

Key Features of Ollama

- **Local Execution:** One of the distinguishing features of Ollama is its ability to run LLMs locally, mitigating privacy concerns associated with cloud-based solutions. By bringing AI models directly to users' devices, Ollama ensures greater control and security over data while providing faster processing speeds and reduced reliance on external servers.
- **Extensive Model Library:** Ollama offers access to an extensive library of pre-trained LLMs, including popular models like Llama 3. Users can choose from a range of models tailored to different

tasks, domains, and hardware capabilities, ensuring flexibility and versatility in their AI projects.

- **Seamless Integration:**Ollama seamlessly integrates with a variety of tools, frameworks, and programming languages, making it easy for developers to incorporate LLMs into their workflows.

Whether it's Python, LangChain, or LlamaIndex, Ollama provides robust integration options for building sophisticated AI applications and solutions.

- **Customization and Fine-tuning:**With Ollama, users have the ability to customize and fine-tune LLMs to suit their specific needs and preferences. From prompt engineering to few-shot learning and fine-tuning processes, Ollama empowers users to shape the behavior and outputs of LLMs, ensuring they align with the desired objectives.

Prerequisites:

- **Computer:**Ollama is currently available for Linux and macOS and windowsoperating systems,For windows it recently preview version is lanched.

- **Basic understanding of command lines:**While Ollama offers a user-friendly interface, some comfort with basic command-line operations is helpful.

Step 1: Download Ollama

- Visit the official Ollama website:<https://ollama.com/>

- Click on the download button corresponding to your operating system (Linux, macOS or Windows (preview)).

- This will download the Ollama installation script.

Step 2: Install Ollama

- Open a terminal window.

- Navigate to the directory where you downloaded the Ollama installation script (usually the Downloads folder).

- Depending on your operating system, use the following commands to grant the script execution permission and then run the installation:

Step 2: Install Ollama

- For linux

Step 2: Install Ollama

- For macOS

Step 2: Install Ollama

- For windowsDirect installations with clicking the downloaded file and follow the on-screen instructions during the installation process
- Direct installations with clicking the downloaded file and follow the on-screen instructions during the installation process

Step 2: Install Ollama

- Direct installations with clicking the downloaded file and follow the on-screen instructions during the installation process

Step 3: Pull Your First Model (Optional)

- Ollama allows you to run various open-source LLMs. Here, we'll use Llama 3 as an example.
- Use the following command to download the Llama 3 model:

Step 3: Pull Your First Model (Optional)

Replace 'gemma' with the specific model name if desired

Step 3: Pull Your First Model (Optional)

The Ollama library curates a diverse collection of LLMs, each with unique strengths and sizes. Some example are as follows:

Step 3: Pull Your First Model (Optional)

- Llama 3 (8B, 70B)
- Phi-3 (3.8B)

- Mistral (7B)
- Neural Chat (7B)
- Starling (7B)
- Code Llama (7B)
- Llama 2 Uncensored (7B)
- LLaVA (7B)
- Gemma (2B, 7B)
- Solar (10.7B)

Step 4: Run and Use the Model

- Once you have a model downloaded, you can run it using the following command:

Step 4: Run and Use the Model

Output for command "ollama run phi3":

Managing Your LLM Ecosystem with the Ollama CLI

The Ollama command-line interface (CLI) provides a range of functionalities to manage your LLM collection:

Managing Your LLM Ecosystem with the Ollama CLI

- Create Models: Craft new models from scratch using the `ollama create` command.
- Pull Pre-Trained Models: Access models from the Ollama library with `ollama pull`.
- Remove Unwanted Models: Free up space by deleting models using `ollama rm`.
- Copy Models: Duplicate existing models for further experimentation with `ollama cp`.
- Interacting with Models: The Power of `ollama run`

Managing Your LLM Ecosystem with the Ollama CLI

The `ollama run` command is your gateway to interacting with any model on your machine. Need a quick summary of a text file? Pass it through an LLM and let it do the work. Ollama even supports

multimodal models that can analyze images alongside text.

Managing Your LLM Ecosystem with the Ollama CLI

We can also use ollama using python code as follows:

Managing Your LLM Ecosystem with the Ollama CLI

Output:

Applications of Ollama

- **Creative Writing and Content Generation:**Writers and content creators can leverage Ollama to overcome writer's block, brainstorm content ideas, and generate diverse and engaging content across different genres and formats.
- **Code Generation and Assistance:**Developers can harness Ollama's capabilities for code generation, explanation, debugging, and documentation, streamlining their development workflows and enhancing the quality of their code.
- **Language Translation and Localization:**Ollama's language understanding and generation capabilities make it an invaluable tool for translation, localization, and multilingual communication, facilitating cross-cultural understanding and global collaboration.
- **Research and Knowledge Discovery:**Researchers and knowledge workers can accelerate their discoveries by using Ollama to analyze, synthesize, and extract insights from vast amounts of information, spanning literature reviews,data analysis,hypothesis generation, and knowledge extraction.
- **Customer Service and Support:**Businesses can deploy intelligent chatbots and virtual assistants powered by Ollama to enhance customer service, automate FAQs, provide personalized product recommendations, and analyze customer feedback for improved satisfaction and engagement.
- **Healthcare and Medical Applications:**In the healthcare industry, Ollama can assist in medical documentation, clinical decision support, patient education, telemedicine, and medical research, ultimately improving patient outcomes and streamlining healthcare delivery.

Ethical Considerations and Responsible AI

While the potential of Ollama is vast and promising, it's essential to address ethical considerations and ensure responsible AI practices. From mitigating bias and ensuring fairness to prioritizing privacy, transparency, and human oversight, developers and organizations must navigate these challenges to harness the full potential of Ollama while minimizing risks and promoting societal benefit.

Conclusion

Ollama's revolutionary approach to natural language understanding heralds a new era where AI can learn and interpret human language as effortlessly as a child does. As researchers continue to refine this innovative model, we stand on the brink of witnessing an unprecedented leap in machine intelligence that could reshape our digital world.

Conclusion

As AI technology continues to evolve, Ollama is poised to play a pivotal role in shaping its future development and deployment. With ongoing advancements in model capabilities, hardware optimization, decentralized model sharing, user experiences, and ethical AI frameworks, Ollama remains at the forefront of AI innovation, driving progress and democratization across all sectors of society.

Conclusion

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Conclusion

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Conclusion

- Artificial Intelligence
- AI-ML-DS
- NLP
- Open Source
- AI-ML-DS With Python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner

- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL

- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development

- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements

- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

RAG(Retrieval-Augmented Generation) using LLama3

-
-

-

RAG(Retrieval-Augmented Generation) using LLama3

RAG, or Retrieval-Augmented Generation, represents a groundbreaking approach in the realm of natural language processing (NLP). By combining the strengths of retrieval and generative models, RAG delivers detailed and accurate responses to user queries. When paired with LLAMA 3, an advanced language model renowned for its nuanced understanding and scalability, RAG achieves new heights of capability. This article explores the synergy between RAG and LLAMA 3, outlining their benefits and providing a step-by-step guide for building a project that leverages these technologies.

What is RAG?

RAG is a framework designed to handle a range of NLP tasks, including question-answering, summarization, and conversational agents. It comprises three main stages:

What is RAG?

- Retrieve: In this initial phase, relevant documents are retrieved from a corpus based on the user's query. Traditional search engines rely on keyword matching, but RAG goes beyond this by utilizing advanced language models for semantic understanding.
- Aggregate: Once documents are retrieved, the next step is to aggregate the information contained within them. This involves condensing and summarizing the content to extract key insights or answers to the user's query.
- Generate: Finally, the aggregated information is used to generate a coherent response or answer. This could involve paraphrasing, synthesizing new information, or providing additional context to enrich the response.

Why RAG Before Fine-Tuning?

Before delving into the implementation details, it's essential to understand why RAG is preferred over fine-tuning in certain scenarios:

Why RAG Before Fine-Tuning?

- Improved Significance and Context: RAG augments the model with pertinent context from a vast corpus of documents, leading to more accurate and contextually relevant responses.
- Enhanced Data Effectiveness: By sparing the model from encoding all external knowledge during pre-training or fine-tuning, RAG allows for the utilization of a large quantity of external knowledge. This dynamic access to information minimizes the need for extensive task-specific data during fine-tuning, making it more effective.
- Effectiveness with Long-Tail Requests: RAG-equipped models excel at handling rare or unseen queries by obtaining relevant data from external sources to fill knowledge gaps. Fine-tuning further enhances the model's performance on less frequent queries by teaching it to utilize retrieved information effectively.

What is LLAMA 3?

LLama3, short for Language Model for Local Document Search, serves as the retrieval component in the RAG framework. Unlike traditional search engines, LLama3 leverages language models to understand the user's query in a more nuanced manner. It retrieves documents from a local corpus based on semantic relevance, rather than relying solely on keyword matching.

Project for Extracting Insights from Documents and URLs

To showcase the integration of RAG with LLAMA 3, we'll build a project using Phidata, a framework that enhances language models' capabilities. By adding memory, knowledge, and tools, Phidata enables language models to engage in more complex interactions and tasks.

Project for Extracting Insights from Documents and URLs

Phidata Framework:

Project for Extracting Insights from Documents and URLs

Phidata is a framework designed for Language Model augmentation. It addresses the limitations of language models by adding the following components:

Project for Extracting Insights from Documents and URLs

- Memory: Allows Language Models to maintain long-term conversations by storing chat history in a database.
- Knowledge: Provides Language Models with contextual information by storing data in a vector database.
- Tools: Enables Language Models to perform actions like pulling data from APIs, sending emails, or querying databases.

Project for Extracting Insights from Documents and URLs

General Steps:

Project for Extracting Insights from Documents and URLs

- Download LLAMA 3: Obtain LLAMA 3 from its official website.
- Clone Phidata Repository: Clone the Phidata Git repository or download the code from the repository.
- Navigate to the RAG Directory: Access the RAG directory within the Phidata repository.
- Set Up Environment: Create a new Python environment using Conda, then install the necessary packages.
- Install Packages: Use Phidata to pull LLAMA 3 and other required packages.
- Database Setup: Install Docker and create a PostgreSQL database using the provided command.
- Run the Application: Launch the application using Streamlit.

Step By Step Guide :

- Step 1- Download the ollama form ollama.com
- Step 2- Clone a Phidata Git repository or take this code

Step By Step Guide :

- Step 3 -Navigate to this path in the repo

Step By Step Guide :

- Step 4 -Setup Environment

Step By Step Guide :

- Step 5 -Install Packages

Step By Step Guide :

- Step 6 -To create a database first install the docker and create the account
- Step 7 -use this command in the same path

Step By Step Guide :

- Step 8 -Run the Application

Step By Step Guide :

Output:

Demonstration of the App

- First add any URL or File

Demonstration of the App

- Ask any question related to the source

Demonstration of the App

- It will take some time and then display the response

Conclusion

To sum up, the use of LLaMA 3 for the Rag task has shown great promise in improving natural language comprehension and generation. LLaMA 3's sophisticated ability to process and generate human-like text when combined with retrieval methods provides greater precision and relevance in the generated content. The combination of retrieval and LLaMA 3 allows the model to gain access to a vast library of knowledge, making sure that responses are contextually relevant and enriched with

relevant information.

Conclusion

"This course is very well structured and easy to learn. Anyone with zero experience of data science, python or ML can learn from this. This course makes things so easy that anybody can learn on their own. It's helping me a lot. Thanks for creating such a great course."-Ayushi Jain | Placed at MicrosoftNow's your chance to unlock high-earning job opportunities as a Data Scientist! Join ourComplete Machine Learning & Data Science Programand get a 360-degree learning experience mentored by industry experts.Get hands on practice with40+ Industry Projects, regular doubt solving sessions, and much more. Register for the Program today!

Conclusion

M

Conclusion

- Blogathon
- Artificial Intelligence
- AI-ML-DS
- Data Science Blogathon 2024
- AI-ML-DS With Python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers

- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Chatbot Theory Explained!

-
-
-

Chatbot Theory Explained!

Chatbots have emerged as a quintessential part of present-day virtual interactions, revolutionizing the way groups interact with customers and customers interact with the era. At the core of chatbot development lies a rich theoretical framework that draws upon diverse disciplines along with artificial intelligence, natural language processing, human-computer interaction, and software program engineering.

Chatbot Theory Explained!

Chatbot

Chatbot Theory Explained!

Understanding the chatbot concept is important for designing, growing, and deploying effective conversational marketers able to know how and respond to consumer queries in natural language.

In this article, chatbots we will discuss every point about chatbots.

What is Chatbot?

A chatbot is a software program for simulating intelligent conversations with humans using rules or artificial intelligence. A computer program that can comprehend human language and communicate with a user via a website or messaging app is known as a chatbot (conversational interface, AI agent). Chatbots are conversational technologies that effectively carry out repetitive activities. They are well-liked by people because they facilitate the speedy completion of those tasks, freeing them up to concentrate on more complex, strategic, and interesting duties that call for human qualities that are unmatched by computers. Online, chatbots can perform a wide range of functions, including setting up calls, responding to basic inquiries, and obtaining client feedback. Bots are used by brands to expedite customer assistance, automate company operations, and reduce support expenses.

The Evolution of Chatbots

The evolution of chatbots represents a charming journey via the annals of AI and human-computer interplay. Over the years, chatbots have undergone a remarkable transition, evolving from its basic text-based programs to sophisticated digital assistants with natural language and context recognition. This history reflects the convergence of scientific discoveries, technology advancements, and practical worldwide programs, and it echoes the unrelenting quest to replicate human-like conversational abilities in machines. Let's discuss evolution of Chatbots.

1. Eliza (1964-1966)

Eliza's creation marked the beginning of a chatbot's existence. Joseph Weizenbaum created Eliza, the first chatbot in history, between 1964 and 1966. Joseph Weizenbaum held the position of professor at MIT. Eliza was designed to employ pattern-matching algorithms to produce a conversation that sounds human. Joseph Weizenbaum, the designer of Eliza, believes that because Eliza was the first artificial intelligence chatbot, it will assist the patient in resolving their

psychological issue. It will undoubtedly aid in their quick recovery in addition to the treatment. Conversely, other people claimed that Eliza was incapable of speaking with true comprehension.

2. Parry (1972)

Parry is regarded as Eliza's improved version. Kenneth Colby, a psychiatrist at Stanford University, invented it in 1972. It might mimic a person suffering from schizophrenia, a mental illness characterised by symptoms including delusions, hallucinations, disorganisation, etc. Using a conversational approach, this chatbot was conveyed to assist those experiencing similar psychological challenges. Parry is thought of as the 'angry Eliza'.

3. The Jabberwacky (1988)

British programmer Rollo Carpenter invented Jabberwacky. Jabberwacky's main objective was to simulate human conversation in a funny, engaging, and entertaining way. It was renowned for its clever and humorous delivery of the text. Pattern-matching is another method that Jabberwacky utilises to interact with people.

4. A. L. I. C. E (1995)

Eliza was the first chatbot, and her success encouraged programmers to create more sophisticated and feature-rich chatbots. The best chatbot, A. L. I. C. E, was awarded the Loebner Prize in 2000, 2001, and 2004. Alicebot is available as open source from the Richard Wallace GitHub account and the ALICE AI foundation on Google Code. The acronym A. L. I. C. E. denotes Artificial Linguistic Internet Computer entity, which is another name for Alicebot. Richard Wallace created A. L. I. C. E., which was published on November 23, 1995.

5. SmarterChild (2001)

The Smarter Child was built using natural language processing, just like any other chatbot. In the year 2001, ActiveBuddy Inc. developed a clever artificial intelligence. The most well-known AI bot that creates humorous online interactions and seems human is called SmarterChild. This AI tool is available on AOL IM and Microsoft Messenger.

6. Google Assistance

Google starts developing Google Assistant on 18 May 2016. Both mobile phones and home automation devices can access it. One can have a two-way discussion with it. You may either speak or text directions to the Google Assistant. Users can even ask Google Assistant to modify system settings and set alarms for reminders. Google, the firm that created the virtual assistant, has revealed that in the near future, the assistant would be able to identify items and things and aid with conducting money transfers and making purchases.

8. ChatGPT

The most sophisticated AI chatbot, ChatGPT, has mostly affected the business world because it can generate text that appears human when interacting with a company's clients or customers. Using information from their profile, past purchases, and the text, ChatGPT assists in creating personalised customer answers.

1. Menu or Button-Based Chatbot

The simplest type of chatbots are menu-based or button-based chatbot, in which users can communicate with them by selecting the button from a scripted menu that most closely matches their requirements. The user-friendly chatbot may present a new set of possibilities based on their clicks, which they can proceed to select until they arrive at the most appropriate and targeted option. These chatbots function essentially like a decision tree.

2. Rule-Based Chatbot

Expanding on the basic decision tree capability of the menu-based chatbot, the rules-based chatbot utilises conditional if/then logic to create automated conversation flows. Rule-based bots function similarly to interactive FAQs, with the conversation designer programming preset question-and-answer combinations into the bot so it can comprehend user input and provide relevant responses.

3. AI Powered Chatbots

AI chatbots can comprehend user questions regardless of how they are expressed, however the conversational flow of rules-based chatbots only allows predefined questions and answer possibilities. The AI bot's natural language understanding (NLU) and artificial intelligence (AI) skills enable it to promptly identify any pertinent contextual information that the user shares, facilitating a more conversational and seamless exchange of ideas. The AI-powered chatbot may pose clarifying questions when it is unclear what a user is requesting and discovers multiple actions that could satisfy the request.

4. Voice Chatbots

A Voice bot is a type of artificial intelligence (AI) software that can converse with incoming calls in contact centres. Using natural language processing (NLP) and machine learning, it records, decodes, and interprets voice input from users and answers intelligibly. Another conversational technology that lets users engage with the bot by speaking to it instead of typing is called voice chatbot. Certain voice chatbots are more basic than others.

Applications of Chatbots

- **Customer Service and Support:** Chatbots are substantially utilized in customer support to provide help and assistance to customers. They can answer frequently asked questions, troubleshoot problems, and guide users via self-care options, thereby reducing wait times and enhancing overall customer experience.
- **E-trade and Retail:** In the e-trade quarter, chatbots are employed to assist buyers with product hints, answer inquiries about product capabilities, pricing, and availability, and facilitate seamless transactions.
- **Financial Services:** Chatbots can help users manipulate their finances, manage prices, set budgets, and even offer investment pointers primarily based on personal desires and chance profiles.
- **Healthcare :** In healthcare, chatbots are applied for a number of functions, including appointment scheduling, medicine reminders, symptom evaluation, and patient schooling. They can triage affected person inquiries, offer fundamental clinical advice, and offer assist for intellectual fitness

and well-being.

- Education and Training: Chatbots are widely used in schooling and education settings to supply customized mastering education, provide tutoring and academic courses.

Benefits of Chatbots

- 24/7 Availability: Chatbots can provide continuous support, improving customer service accessibility.
- Scalability: They can take care of a couple of conversations concurrently, scaling effortlessly to accommodate growing demand.
- Cost Savings: Automation reduces the need for human agents, resulting in fee savings for agencies.
- Instant Responses: Chatbots deliver immediate responses, improving reaction times and person delight.
- Consistency: They provide regular and standardized responses, ensuring uniformity in customer interactions.
- Data Collection: Chatbots can acquire precious consumer statistics and insights, enabling personalized stories and centered advertising and marketing.
- Integration: Chatbots can combine with present systems and structures, enhancing workflow efficiency.

Limitations of Chatbots

- Lack of Understanding: Chatbots may also struggle to recognize complicated queries or nuances in language, which leads to misunderstandings.
- Limited Scope: They are powerful as their programmed abilities, restricting their ability to handle unforeseen data.
- Impersonal Interactions: Some customers may also decide on human interplay over interacting with a system, leading to dissatisfaction.
- Technical Issues: Chatbots are prone to technical system faults and errors, that can disrupt consumer reports.

- **Dependency:**Overreliance on chatbots may result in reduced human interaction and lack of interpersonal connections.
- **Initial Investment:**Developing and imposing chatbots requires preliminary investment in phrases of time, resources, and technical knowledge.
- **Maintenance:**Chatbots require ongoing protection and updates to stay powerful and relevant through the years.

Conclusion

In conclusion, chatbots have come to be indispensable tools for corporations in search of enhance customer service, automate obligations, and benefit. As chatbot generation evolves, advancements in artificial intelligence and natural language processing are enhancing their talents, enabling more user satisfaction. While demanding situations such as understanding complex queries and privacy issues continue to be, ongoing innovation and refinement are addressing those issues and improving the general effectiveness of chatbots. By leveraging the strengths of chatbot technology and addressing its barriers, agencies can release new opportunities for growth, efficiency, and innovation inside the digital age.

What is Chatbot?

A chatbot is a software program for simulating intelligent conversations with humans using rules or artificial intelligence.

Where are chatbots used?

Chatbots are widely used in business process, customer service, finance and in healthcare.

What are some examples of successful chatbots?

Some example of chatbots are Alexa, Siri, Google Assistance etc.

What is the future of chatbots?

Potentially reshaping the future through task simplification and improved consumer experiences are

chatbots. They will also automate procedures in a variety of sectors. This change will encourage innovation and increase efficiency.

What are the benefits of using chatbots?

- Chatbots can provide continuous support, improving customer service accessibility.
- They can take care of a couple of conversations concurrently, scaling effortlessly to accommodate growing demand.
- Automation reduces the need for human agents, resulting in fee savings for agencies.
- Chatbots deliver immediate responses, improving reaction times and person delight.

What are the benefits of using chatbots?

Get 90% Course fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today. The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

What are the benefits of using chatbots?

- GBlog
- GBlog 2024

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media

- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS

- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git

- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter

- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course

- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification

- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Classification Projects

- Wine Quality Prediction - Machine Learning
- ML | Credit Card Fraud Detection
- Disease Prediction Using Machine Learning
- Recommendation System in Python
- Detecting Spam Emails Using Tensorflow in Python
- SMS Spam Detection using TensorFlow in Python
- Python | Classify Handwritten Digits with Tensorflow
- Recognizing HandWritten Digits in Scikit Learn
- Identifying handwritten digits using Logistic Regression in PyTorch
- Python | Customer Churn Analysis Prediction
- Online Payment Fraud Detection using Machine Learning in Python
- Flipkart Reviews Sentiment Analysis using Python
- Loan Approval Prediction using Machine Learning
- Loan Eligibility Prediction using Machine Learning Models in Python
- Stock Price Prediction using Machine Learning in Python
- Bitcoin Price Prediction using Machine Learning in Python

- Handwritten Digit Recognition using Neural Network
- Parkinson Disease Prediction using Machine Learning - Python
- Spaceship Titanic Project using Machine Learning - Python
- Rainfall Prediction using Machine Learning - Python
- Autism Prediction using Machine Learning
- Predicting Stock Price Direction using Support Vector Machines
- Fake News Detection Model using TensorFlow in Python
- CIFAR-10 Image Classification in TensorFlow
- Black and white image colorization with OpenCV and Deep Learning
- ML | Kaggle Breast Cancer Wisconsin Diagnosis using Logistic Regression
- ML | Cancer cell classification using Scikit-learn
- ML | Kaggle Breast Cancer Wisconsin Diagnosis using KNN and Cross Validation
- Human Scream Detection and Analysis for Controlling Crime Rate - Project Idea
- Multiclass image classification using Transfer learning
- Intrusion Detection System Using Machine Learning Algorithms
- Heart Disease Prediction using ANN

Regression Projects

- IPL Score Prediction using Deep Learning
- Dogecoin Price Prediction with Machine Learning
- Zillow Home Value (Zestimate) Prediction in ML
- Calories Burnt Prediction using Machine Learning
- Vehicle Count Prediction From Sensor Data
- Analyzing selling price of used cars using Python
- Box Office Revenue Prediction Using Linear Regression in ML
- House Price Prediction using Machine Learning in Python
- ML | Boston Housing Kaggle Challenge with Linear Regression

- Stock Price Prediction Project using TensorFlow
- Medical Insurance Price Prediction using Machine Learning - Python
- Inventory Demand Forecasting using Machine Learning - Python
- Ola Bike Ride Request Forecast using ML
- Waiter's Tip Prediction using Machine Learning
- Predict Fuel Efficiency Using Tensorflow in Python
- Microsoft Stock Price Prediction with Machine Learning
- Share Price Forecasting Using Facebook Prophet
- Python | Implementation of Movie Recommender System
- How can Tensorflow be used with abalone dataset to build a sequential model?

Computer Vision Projects

- OCR of Handwritten digits | OpenCV
- Cartooning an Image using OpenCV - Python
- Count number of Object using Python-OpenCV
- Count number of Faces using Python - OpenCV
- Text Detection and Extraction using OpenCV and OCR
- FaceMask Detection using TensorFlow in Python
- Dog Breed Classification using Transfer Learning
- Flower Recognition Using Convolutional Neural Network
- Emojify using Face Recognition with Machine Learning
- Cat & Dog Classification using Convolutional Neural Network in Python
- Traffic Signs Recognition using CNN and Keras in Python
- Lung Cancer Detection using Convolutional Neural Network (CNN)
- Lung Cancer Detection Using Transfer Learning
- Pneumonia Detection using Deep Learning
- Detecting Covid-19 with Chest X-ray

- Skin Cancer Detection using TensorFlow
- Age Detection using Deep Learning in OpenCV
- Face and Hand Landmarks Detection using Python - Mediapipe, OpenCV
- Detecting COVID-19 From Chest X-Ray Images using CNN
- Image Segmentation Using TensorFlow
- License Plate Recognition with OpenCV and Tesseract OCR
- Detect and Recognize Car License Plate from a video in real time
- Residual Networks (ResNet) - Deep Learning

Natural Language Processing Projects

- Twitter Sentiment Analysis using Python
- Facebook Sentiment Analysis using python
- Next Sentence Prediction using BERT
- Hate Speech Detection using Deep Learning
- Image Caption Generator using Deep Learning on Flickr8K dataset
- Movie recommendation based on emotion in Python
- Speech Recognition in Python using Google Speech API
- Voice Assistant using python
- Human Activity Recognition - Using Deep Learning Model
- Fine-tuning BERT model for Sentiment Analysis
- Sentiment Classification Using BERT
- Sentiment Analysis with an Recurrent Neural Networks (RNN)
- Autocorrector Feature Using NLP In Python
- Python | NLP analysis of Restaurant reviews
- Restaurant Review Analysis Using NLP and SQLite
- Twitter Sentiment Analysis using Python

Clustering Projects

- Customer Segmentation using Unsupervised Machine Learning in Python
- Music Recommendation System Using Machine Learning
- K means Clustering - Introduction
- Image Segmentation using K Means Clustering

Recommender System Project

- AI Driven Snake Game using Deep Q Learning

Recommendation System in Python

-
-
-

Recommendation System in Python

The ability to generate value for businesses by leveraging data and applying pertinent programming abilities is the fundamental component of both Data Science (DS) and Artificial Intelligence (AI). The way individuals can now access and enjoy products and services from the comfort of their homes with just a few clicks has been transformed by industry leaders like Netflix, Amazon, and Uber Eats. These platforms have used recommendation algorithms to improve the user experience. Users are catered to by these systems, which provide an abundance of customized options that are carefully crafted to suit their individual interests and tastes. Within this framework, Python is a vital resource that provides an adaptable and strong environment for creating and implementing state-of-the-art recommendation systems. There are a lot of applications where websites collect data from their users and use that data to predict the likes and dislikes of their users. This enables people to suggest the material that interests them. Recommender systems are a means of making suggestions for products and concepts that align with a user's particular perspective.

Recommendation System in Python

Python Recommendation System employs a data-driven methodology to offer customers tailored

recommendations. It uses user data and algorithms to forecast and suggest goods, services, or content that a user is probably going to find interesting. These systems are essential in applications where users may become overwhelmed by large volumes of information, such as social media, streaming services, and e-commerce. Building recommendation systems is a common use for Python because of its modules and machine learning frameworks. The two main kinds are content-based filtering (which takes into account the characteristics of products and user profiles) and collaborative filtering (which generates recommendations based on user behaviour and preferences). Hybrid strategies that integrate the two approaches are also popular. These kinds of systems improve user experiences, boost user involvement, and propel corporate expansion.

Recommendation System in Python

Recommender System is of different types:

Recommendation System in Python

- Content-Based Recommendation: It is supervised machine learning used to induce a classifier to discriminate between interesting and uninteresting items for the user.
- Collaborative Filtering: Collaborative Filtering recommends items based on similarity measures between users and/or items. The basic assumption behind the algorithm is that users with similar interests have common preferences.

Content-Based Recommendation System

Content-based systems recommend items to the customer similar to previously high-rated items by the customer. It uses the features and properties of the item. From these properties, it can calculate the similarity between the items.

Content-Based Recommendation System

In a content-based recommendation system, first, we need to create a profile for each item, which represents the properties of those items. The user profiles are inferred for a particular user. We use these user profiles to recommend the items to the users from the catalog.

Item profile

In a content-based recommendation system, we need to build a profile for each item, which contains the important properties of each item. For Example, If the movie is an item, then its actors, director, release year, and genre are its important properties, and for the document, the important property is the type of content and set of important words in it.

Item profile

Let's have a look at how to create an item profile. First, we need to perform the TF-IDF vectorizer, here TF (term frequency) of a word is the number of times it appears in a document and The IDF (inverse document frequency) of a word is the measure of how significant that term is in the whole corpus.

TF-IDF Vectorizer

- Term Frequency(TF): Term frequency, or TF for short, is a key idea in information retrieval and natural language processing. It displays the regularity with which a certain term or word occurs in a text corpus or document. TF is used to rank terms in a document according to their relative value or significance. The term-frequency can be calculated by:
$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$
 where f_{ij} is the frequency of term (feature) i in document (item) j . For a variety of text analysis tasks, such as information retrieval, document classification, and sentiment analysis, the yielded TF value can be used to identify important terms in a document. It offers a framework for figuring out how relevant a word is in a particular situation.

- Inverse-document Frequency(IDF): The measure known as Inverse Document Frequency (IDF) is employed in text analysis and information retrieval to evaluate the significance of phrases within a set of documents. IDF measures how uncommon or unique a term is in the corpus. To compute it, take the reciprocal of the fraction of documents that include the term and logarithmize it. Common terms have lower IDF values, while rare terms have higher values. IDF is an essential part of the TF-IDF (Term Frequency-Inverse Document Frequency) method, which uses it to assess the relative importance of terms in different documents. To improve information representation and

retrieval from massive text datasets, IDF is used in tasks including document ranking, categorization, and text mining. The inverse-document frequency can be calculated with: $IDF_{\{i\}} = \log_e \frac{N}{n_i}$ where, n_i is the number of documents that mention term i . N is the total number of docs.

TF-IDF Vectorizer

A numerical statistic called Term Frequency-Inverse Document Frequency (TF-IDF) is employed in information retrieval and natural language processing. The term's significance within a document is assessed in relation to a group of documents (the corpus). TF emphasizes terms with greater frequencies by measuring a term's frequency of occurrence in a document. IDF evaluates a term's rarity within the corpus, emphasizing terms that are distinct. A weighted score is produced for each term in a document by multiplying TF and IDF together to compute TF-IDF.

TF-IDF Vectorizer

Therefore, the total formula is:

User profile

The user profile is a vector that describes the user preference. During the creation of the user's profile, we use a utility matrix that describes the relationship between user and item. From this information, the best estimate we can decide which item the user likes, is some aggregation of the profiles of those items.

User profile

Advantages and Disadvantages

User profile

- Advantages: No need for data on other users when applying to similar users. Able to recommend to users with unique tastes. Able to recommend new & popular items. Explanations for recommended items.

- No need for data on other users when applying to similar users.
- Able to recommend to users with unique tastes.
- Able to recommend new & popular items
- Explanations for recommended items.
- Disadvantages: Finding the appropriate feature is hard. Doesn't recommend items outside the user profile.
- Finding the appropriate feature is hard.
- Doesn't recommend items outside the user profile.

User profile

- No need for data on other users when applying to similar users.
- Able to recommend to users with unique tastes.
- Able to recommend new & popular items
- Explanations for recommended items.

User profile

- Finding the appropriate feature is hard.
- Doesn't recommend items outside the user profile.

Collaborative Filtering

Collaborative filtering is based on the idea that similar people (based on the data) generally tend to like similar things. It predicts which item a user will like based on the item preferences of other similar users.

Collaborative Filtering

Collaborative filtering uses a user-item matrix to generate recommendations. This matrix contains the values that indicate a user's preference towards a given item. These values can represent either explicit feedback (direct user ratings) or implicit feedback (indirect user behavior such as listening, purchasing, watching).

Collaborative Filtering

- Explicit Feedback: The amount of data that is collected from the users when they choose to do so. Many of the times, users choose not to provide data for the user. So, this data is scarce and sometimes costs money. For example, ratings from the user.
- Implicit Feedback: In implicit feedback, we track user behavior to predict their preference.

Collaborative Filtering

Example:

Collaborative Filtering

- Consider a user x , we need to find another user whose rating are similar to x 's rating, and then we estimate x 's rating based on another user.

Collaborative Filtering

| M_1 | M_2 | M_3 | M_4 | M_5 | M_6 | M_7

A | 4 | | | 5 | 1 | |

B | 5 | 5 | 4 | | | 5 |

C | | | | 2 | 4 | |

D | | 3 | | | | 3

Collaborative Filtering

- Let's create a matrix representing different user and movies:
- Consider two users x , y with rating vectors r_x and r_y . We need to decide a similarity matrix to calculate similarity b/w $\text{sim}(x, y)$. There are many methods to calculate similarity such as: Jaccard similarity, cosine similarity and pearson similarity. Here, we use centered cosine similarity/ pearson similarity, where we normalize the rating by subtracting the mean:

Collaborative Filtering

| M_1 | M_2 | M_3 | M_4 | M_5 | M_6 | M_7

A | 2/3 | | | 5/3 | -7/3 | |

B | 1/3 | 1/3 | -2/3 | | | |

C | | | | -5/3 | 1/3 | 4/3 |

D | | 0 | | | | 0

Collaborative Filtering

- Here, we can calculate similarity: For ex: $\text{sim}(A,B) = \cos(r_A, r_B) = 0.09$; $\text{sim}(A,C) = -0.56$. $\text{sim}(A,B) > \text{sim}(A,C)$.

Rating Predictions

Let r_x be the vector of user x 's rating. Let N be the set of k similar users who also rated item i . Then we can calculate the prediction of user x and item i by using following formula:

Advantages and Disadvantages

- Advantages: No need for the domain knowledge because embedding are learned automatically. Capture inherent subtle characteristics.
- No need for the domain knowledge because embedding are learned automatically.
- Capture inherent subtle characteristics.
- Disadvantages: Cannot handle fresh items due to cold start problem. Hard to add any new features that may improve quality of model
- Cannot handle fresh items due to cold start problem.
- Hard to add any new features that may improve quality of model

Advantages and Disadvantages

- No need for the domain knowledge because embedding are learned automatically.
- Capture inherent subtle characteristics.

Advantages and Disadvantages

- Cannot handle fresh items due to cold start problem.

- Hard to add any new features that may improve quality of model

Importing Libraries

The Python environment for data analysis and visualization is initialized using this line of code. First, it imports essential libraries for data processing and visualization, including NumPy, Pandas, scikit-learn, Matplotlib, and Seaborn. It also sets up the code to suppress future warnings, so that cautions about upcoming library changes don't clog the output and create a messier, less productive workspace. These preparatory actions create the framework for effective data exploration and analysis with the imported tools.

Loading Datasets

Output:

Loading Datasets

```
userId  movieId  rating  timestamp0      1      1    4.0 9649827031      1      3    4.0 9649812472
      1      6    4.0 9649822243      1     47    5.0 9649838154      1     50    5.0 964982931
```

Loading Datasets

Output:

Loading Datasets

```
movieId          title \0      1          Toy Story (1995)1      2          Jumanji
(1995)2      3          Grumpier Old Men (1995)3      4          Waiting to Exhale (1995)4      5
Father of the Bride Part II (1995)genres0  Adventure|Animation|Children|Comedy|Fantasy1
      Adventure|Children|Fantasy2          Comedy|Romance3
Comedy|Drama|Romance4          Comedy
```

Loading Datasets

Two datasets are imported into this code to do a movie recommendation study. User ratings for movies are included in the first dataset, 'ratings.csv', which is kept in a Pandas DataFrame named

ratings. The second dataset, called 'movies.csv', is put into a Pandas DataFrame called 'movies' and contains movie metadata like names and genres. In order to give a preliminary overview of the data and lay the groundwork for further analysis or recommendation system development, the code displays the first few rows of each DataFrame.

Statistical Analysis of Ratings

Output:

Statistical Analysis of Ratings

Number of ratings: 100836
Number of unique movie IDs: 9724
Number of unique users: 610
Average ratings per user: 165.3
Average ratings per movie: 10.37

Statistical Analysis of Ratings

This code computes and reports a number of crucial statistics for a movie ratings dataset. It counts the number of unique movie IDs (n_movies) and user IDs (n_users) as well as the total number of ratings (n_ratings). These metrics provide important information about the properties of the dataset, including its size and the variety of people and movies inside it. To give a more complete picture of the distribution of ratings throughout the dataset, it also calculates and shows the average number of ratings for each user and each movie. Understanding the size and user interaction of the dataset requires knowledge of this information.

User Rating Frequency

Output:

User Rating Frequency

userId	n_ratings	0	1	2321	2	292	3	393	4	2164	5	44
--------	-----------	---	---	------	---	-----	---	-----	---	------	---	----

User Rating Frequency

The movie ratings dataset's user-specific statistics are computed and shown in this code segment. After classifying the data according to user IDs, it calculates the total number of ratings each user

has submitted and saves the results in a new DataFrame named `user_freq`. With `userId` denoting the user ID and `n_ratings` the number of ratings the user has contributed, the columns are suitably labeled. To facilitate additional user-based analysis and the creation of recommendation systems, this user-level frequency information is crucial for comprehending user engagement and activity inside the rating dataset. The first few rows of this DataFrame are shown for a brief summary of user-specific rating counts by the `print(user_freq.head())` line.

Movie Rating Analysis

To determine which movies in the dataset have the lowest and highest ratings, this algorithm analyzes movie reviews. It determines the average ratings for every film, making it possible to identify which ones have the lowest and greatest average ratings. Subsequently, the algorithm accesses and presents the information about these films from the `movies` dataset. It also sheds light on the popularity and audience involvement of the movie by displaying the number of users who rated both the highest and lowest-ranked ones. This gives insights into user engagement. Bayesian averages may offer more accurate quality ratings for films with a small number of ratings.

User-Item Matrix Creation

A user-item matrix is a basic data structure in recommendation systems, and it is created by the code that is given. This is how it operates:

User-Item Matrix Creation

- To find the number of unique users and unique videos in the dataset, `N` and `M` are computed.
- There are four dictionaries produced:
 - `user_mapper`: Maps distinct user IDs to indexes (user ID 1 becomes index 0 for example).
 - `movie_mapper`: Converts distinct movie IDs into indices (movie ID 1 becomes index 0 for example).
 - `user_inv_mapper`: Reverses `user_mapper` and maps indices back to user IDs.
 - `movie_inv_mapper`: Reverses `movie_mapper` by mapping indices to movie IDs.
- `user_mapper`: Maps distinct user IDs to indexes (user ID 1 becomes index 0 for example).
- `movie_mapper`: Converts distinct movie IDs into indices (movie ID 1 becomes index 0 for

example).

- user_inv_mapper: Reverses user_mapper and maps indices back to user IDs.
- movie_inv_mapper: Reverses movie_mapper by mapping indices to movie IDs.
- To map the real user and movie IDs in the dataset to their matching indices, the lists user_index and movie_index are generated.
- A sparse matrix X is created using the SciPy function csr_matrix. The user and movie indices that correspond to the rating values in the dataset are used to generate this matrix. The form of it is (M, N), where M denotes the quantity of distinct films and N denotes the quantity of distinct consumers.

User-Item Matrix Creation

- user_mapper: Maps distinct user IDs to indexes (user ID 1 becomes index 0 for example).
- movie_mapper: Converts distinct movie IDs into indices (movie ID 1 becomes index 0 for example).
- user_inv_mapper: Reverses user_mapper and maps indices back to user IDs.
- movie_inv_mapper: Reverses movie_mapper by mapping indices to movie IDs.

User-Item Matrix Creation

To put it another way, this code makes it easy to do calculations and create recommendation systems based on the structured representation of user ratings for movies in the data.

Movie Similarity Analysis

Output:

Movie Similarity Analysis

Since you watched Grumpier Old Men (1995)Grumpy Old Men (1993)Striptease (1996)Nutt
y Professor, The (1996)Twister (1996)Father of the Bride Part II (1995)Broken Arrow (1996)Bio-Dome
(1996)Truth About Cats & Dogs, The (1996)Sabrina (1995)Birdcage, The (1996)

Movie Similarity Analysis

The provided code defines a function, `find_similar_movies`, which uses the k-Nearest Neighbors (KNN) algorithm to identify movies that are similar to a given movie. The function takes inputs such as the target movie ID, a user-item matrix (X), the number of neighbors to consider (k), a similarity metric (default is cosine similarity), and an option to show distances between movies. The function begins by initializing a blank list to hold the IDs of films that are comparable. It takes the target movie's index out of the `movie_mapper` dictionary and uses the user-item matrix to acquire the feature vector that goes with it. Next, the KNN model is configured using the given parameters.

Movie Similarity Analysis

The distances and indices of the k-nearest neighbors to the target movie are calculated once the KNN model has been fitted. Using the `movie_inv_mapper` dictionary, the loop retrieves these neighbor indices and maps them back to movie IDs. Since it matches the desired movie, the first item in the list is eliminated. The code ends with a list of related movie titles and the title of the target film, suggesting movies based on the KNN model.

Movie Recommendation with respect to Users Preference

Create a function to recommend the movies based on the user preferences.

Movie Recommendation with respect to Users Preference

The function accepts the following inputs: dictionaries (`user_mapper`, `movie_mapper`, and `movie_inv_mapper`) for mapping user and movie IDs to matrix indices; the `user_id` for which recommendations are desired; a user-item matrix X representing movie ratings; and an optional parameter k for the number of recommended movies (default is 10).

Movie Recommendation with respect to Users Preference

- It initially filters the ratings dataset to see if the user with the given ID is there. It notifies the user that the requested person does not exist and ends the function if the user does not exist (the filtered DataFrame is empty).
- The code, if it exists, designates the movie that has received the highest rating from that particular

user. It finds the movie of this movie and chooses it based on the highest rating.

- With information from the movies dataset, a dictionary called `movie_titles` is created to map movie IDs to their titles. The function then uses `find_similar_movies` to locate films that are comparable to the movie in the user-item matrix that has the highest rating (denoted by `movie_id`). It gives back a list of comparable movie IDs.

- The code searches the movie titles dictionary for the title of the highest-rated film, and if the film is not found, it sets the title to "Movie not found." When a movie title is retrieved as "Movie not found," it means that the highest-rated film (based on `movie_id`) is not present in the dataset. If the movie is located, the customer is presented with recommendations for other movies based on the highest rated film. The list of comparable movie IDs is iterated over, and the titles are printed. When a movie isn't discovered in the dataset, the default message is "Movie not found."

- The function handles situations where the user or movie doesn't exist in the dataset and is intended to suggest movies for a particular user based on their highest-rated film. The code calls the function with the necessary parameters and sets the `user_id` to a specific user to show how to utilize the method.

Recommend the movies

Output:

Recommend the movies

Since you watched Twelve Monkeys (a.k.a. 12 Monkeys) (1995), you might also like: Pulp Fiction (1994) Terminator 2: Judgment Day (1991) Independence Day (a.k.a. ID4) (1996) Seven (a.k.a. Se7en) (1995) Fargo (1996) Fugitive, The (1993) Usual Suspects, The (1995) Jurassic Park (1993) Star Wars: Episode IV - A New Hope (1977) Heat (1995)

Recommend the movies

Output:

Recommend the movies

Since you watched Pulp Fiction (1994), you might also like: Silence of the Lambs, The (1991) Shawshank Redemption, The (1994) Seven (a.k.a. Se7en) (1995) Forrest Gump (1994) Usual Suspects, The (1995) Braveheart (1995) Fight Club (1999) Fargo (1996) Terminator 2: Judgment Day (1991) Reservoir Dogs (1992)

Conclusion

In conclusion, developing a Python recommendation system allows for the creation of tailored content recommendations that improve user experience and take into account user preferences. Through the utilization of collaborative filtering, content-based filtering, and hybrid techniques, these systems are able to offer customized recommendations to consumers for content, movies, or items. These systems use sophisticated methods such as closest neighbors and matrix factorization to find hidden patterns in item attributes and user behavior. Recommendation systems are able to adjust and get better over time thanks to the combination of machine learning and data-driven insights. In the end, these solutions are essential for raising consumer satisfaction, improving user engagement, and propelling corporate expansion in a variety of industries.

Get the Complete Notebook:

Notebook: [Click here](#).

Get the Complete Notebook:

Dataset: [Click Here](#).

Get the Complete Notebook:

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Get the Complete Notebook:

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Get the Complete Notebook:

P

Get the Complete Notebook:

- AI-ML-DS
- Machine Learning
- AI-ML-DS With Python

Get the Complete Notebook:

- Machine Learning

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest

- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Fundamental of Data Science

- What is Data Science?
- What Are the Roles and Responsibilities of a Data Scientist?
- Top 10 Data Science Job Profiles
- Applications of Data Science
- Data Science vs Data Analytics
- Data Science Vs Machine Learning : Key Differences
- Difference Between Data Science and Business Intelligence
- Data Science Fundamentals
- Data Science Lifecycle
- How Much Math Do You Need to Become a Data Scientist?

Programming Language for Data Science

- Python for Data Science - Learn the Uses of Python in Data Science
- R Programming for Data Science
- SQL for Data Science

Complete Data Science Program

- Data Science Tutorial
- Learn Data Science Tutorial With Python

Data Analysis tutorial

- Data Analysis (Analytics) Tutorial
- Data Analysis with Python
- Data analysis using R
- Top 80+ Data Analyst Interview Questions and Answers

Data Vizualazation Tutotrial

- Python - Data visualization tutorial
- Data Visualization with Python

- Getting started with Data Visualization in R

Machine Learning Tutorial

- Machine Learning Tutorial
- Maths for Machine Learning
- 100+ Machine Learning Projects with Source Code [2025]
- Top 50+ Machine Learning Interview Questions and Answers
- Machine Learning with R

Deep Learning & NLP Tutorial

- Deep Learning Tutorial
- 5 Deep Learning Project Ideas for Beginners
- Deep Learning Interview Questions
- Natural Language Processing (NLP) Tutorial
- Top 50 NLP Interview Questions and Answers 2024 Updated

Computer Vision Tutorial

- Computer Vision Tutorial
- 50+ Top Computer Vision Projects [2025 Updated]

Data Science Tutorial

-
-
-

Data Science Tutorial

Data Science is an interdisciplinary field that combines powerful techniques from statistics, artificial intelligence, machine learning, and data visualization to extract meaningful insights from vast amounts of data. It is the technology behind modern decision-making, empowering industries to

optimize operations, predict trends, and drive innovation.

Data Science Tutorial

This Data Science tutorial offers a comprehensive guide to all major concepts, tools, and techniques in data science life cycle, along with real-world examples and projects to help you upskill and become a data scientist.

Prerequisites & Tools for Data Science

To gain expertise in the field of data science, you need to have a strong foundation in various aspects of data science that include following languages:

Prerequisites & Tools for Data Science

- Python for Data Science
- R for Data Science
- SQL for Data Science

Mathematics for Data Science

- Statistics for Data Science
- Linear Algebra for Data Science
- Calculus for Data Science

Mathematics for Data Science

For a detailed overview, you can refer to: [Mathematics for Data Science](#)

Data Loading

- Import Excel file with Pandas
- Import Text file with Pandas
- Read JSON Files with Pandas

Data Preprocessing

- What is Data Preprocessing?
- What is Data Cleaning?
- Checking for missing values
- Handling Missing Data
- Removing duplicates using `drop_duplicates()`
- Handling outliers
- Correcting data types
- Renaming columns in Pandas DataFrame
- Data Transformation using Feature Scaling
- Encoding Categorical Variables
- Data Aggregation and Grouping
- Splitting Data into Training and Testing Sets
- Feature Selection

Data Preprocessing

We can also perform Preprocessing in R programming: Data Preprocessing in R

Data Analysis

- What is Data Analysis?
- Data Analysis Process
- What is Exploratory Data Analysis?
- Exploratory Data Analysis (EDA) using Python
- Understanding dataset using `info()`
- Understanding dataset using `describe()`
- Identifying correlations between features
- Statistical Analysis in Python Hypothesis Testing in Python T-test in Python Z-test in Python ANOVA (Analysis of Variance) in Python Mann-Whitney U Test in Python Shapiro-Wilk Test in Python Wilcoxon Signed-Rank Test in Python

- Hypothesis Testing in Python
- T-test in Python
- Z-test in Python
- ANOVA (Analysis of Variance) in Python
- Mann-Whitney U Test in Python
- Shapiro-Wilk Test in Python
- Wilcoxon Signed-Rank Test in Python

Data Analysis

- Hypothesis Testing in Python
- T-test in Python
- Z-test in Python
- ANOVA (Analysis of Variance) in Python
- Mann-Whitney U Test in Python
- Shapiro-Wilk Test in Python
- Wilcoxon Signed-Rank Test in Python

Data Analysis

We can also perform data analysis in R programming: Data Analysis in R

Data Visualization

1. Visualization with Matplotlib

Data Visualization

- Style Plots using Matplotlib
- Line chart using Matplotlib
- Bar Plot using Matplotlib
- Box Plot using Matplotlib
- Scatter Plot using Matplotlib

- Heatmap using Matplotlib
- Three-dimensional Plotting using Matplotlib

Data Visualization

2.Visualization using Seaborn

Data Visualization

- Pairplot using Seaborn
- Violin Plot using Seaborn
- KDE Plot for distribution visualization

Data Visualization

3.Interactive Visualization using Plotly

Data Visualization

- Scatter Plot using Plotly
- Bar Chart using Plotly
- Line Chart using Plotly
- Animated Data Visualization using Plotly
- Choropleth Maps using Plotly

Data Visualization

- Interactive Visualization using Bokeh
- Visualizing Geospatial Data using Folium

Data Visualization

We can also perform data visualization in R programming:Data Visualization in R

Machine Learning

- What is Machine Learning?

- Types of Machine learning
- Supervised Machine Learning
- Unsupervised Machine Learning
- Reinforcement Learning

Machine learning Algorithms

- Linear Regression
- Logistic Regression
- Decision Trees
- Random Forest
- Support Vector Machines (SVM)
- K-Nearest Neighbors (KNN)
- Naive Bayes
- K-Means Clustering
- Principal Component Analysis (PCA)
- XGBoost

Machine learning Algorithms

The list of machine learning algorithms don't end here, to explore more: [Machine Learning Algorithms](#)

Model Evaluation Techniques

- Confusion Matrix
- Precision, Recall, and F1-Score
- ROC Curve and AUC
- Cross-validation
- Hyperparameter Tuning

Deep Learning

- Artificial Neural Networks (ANN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

Deep Learning

Having mastered the fundamental concepts of data science, you are now ready to explore real-world projects. For detailed guidance and project ideas, refer to the article to take your skills to the next level:

Data Science Projects [With Source code]

Step-by-step Guide to Learn Data Science

Data Science is a field that involves extracting insights and knowledge from data using various techniques and tools. If you are a beginner in Data Science, here are some steps you can follow to get started:

Step-by-step Guide to Learn Data Science

- Learn Programming: Start with Python, the most widely used language in Data Science, and explore libraries like NumPy, Pandas, and Scikit-learn.
- Learn Statistics: Statistics helps to analyze and interpret the data. Start by mastering the basics of statistics to build your analytical capabilities.
- Learn Data Visualization: Data Visualization understands data and identify trends and patterns. Python libraries like Matplotlib and Seaborn are widely used for creating insightful visualizations.
- Learn Machine Learning: ML involves building models that can learn from data and make predictions. Key types of machine learning include supervised learning, unsupervised learning, and reinforcement learning. Begin by learning the basics of machine learning algorithms, model evaluation, and tuning.
- Practice with Projects: Apply your skills through hands-on projects like data cleaning, analysis, and machine learning models. Platforms like Kaggle offer great practice opportunities.

- Learn from the Community: Join online communities such as Geeks for Geeks, Reddit and LinkedIn, and attend Data Science meetups to learn from others and stay updated.
- Continuously Learn: Stay updated with new techniques and tools by reading blogs, research papers, and participating in online courses.

Need for Data Science

There are 4 major reasons why there is a need for data science in the existing world today.

Need for Data Science

- Today, businesses rely heavily on customer insights and data science provide these insights by data mining and sorting techniques. A business can analyze the customer behavior and preferences, allowing them to target areas of interest.

Need for Data Science

- Data Science enables businesses to process and organize large volumes of unstructured data quickly. By using advanced data cleaning and transformation techniques, Data Science helps companies save time and resources while converting raw data into valuable insights.

Need for Data Science

- Data Science helps businesses define their objectives and stay on track toward achieving them. By leveraging predictive analytics, organizations can forecast future trends and make data-driven decisions based on historical behavior patterns.

Need for Data Science

- Data Science empowers organizations by improving recruitment and workforce management. Through data-driven analysis, companies can identify the best candidates, filter out unsuitable applicants, and streamline hiring processes, resulting in better hires and significant time savings.

Careers in Data Science

Data Science has been considered one of the most desirable jobs in the IT field today. The growth

opportunities in data science jobs are comparatively high than in any other job. Companies are now focusing more on data science jobs to elevate their business goals which has also created a flood of data science jobs in the market.

Careers in Data Science

Some of the most notable jobs in data science are:-

Careers in Data Science

- Data Scientist
- Data Architect
- Data Administrator
- Data Analyst
- Business Analyst

Applications of Data Science

There are many applications of data science are as follows:-

Applications of Data Science

- Search Engines,
- Transport, Finance,
- E-Commerce,
- Health Care,
- Image Recognition,
- Targeting recommendations, etc.

What is data science?

Data science is a field that involves using techniques from statistics, mathematics, and computer science to analyze and draw insights from data.

What skills do I need to be a data scientist?

Data scientists typically need skills in statistics, machine learning, data visualization, and programming. Strong communication and critical thinking skills are also important.

What programming languages should I learn for data science?

Some popular programming languages for data science include Python, R, and SQL. It's also helpful to have some familiarity with other languages like Java and C++.

How long does it take to learn data science?

Learning data science is an ongoing process that can take several months to several years, depending on your background and level of experience.

What kind of jobs can I get with a background in data science?

Some common job titles in data science include data analyst, data scientist, machine learning engineer, and business intelligence analyst.

What kind of jobs can I get with a background in data science?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

What kind of jobs can I get with a background in data science?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

What kind of jobs can I get with a background in data science?

- AI-ML-DS

- Data Science

Similar Reads

Similar Reads

- Company

- [About Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)

- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies

- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design

- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry

- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles

- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools

- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]

- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Prerequisites for Data Analysis

- Exploratory Data Analysis (EDA) with NumPy, Pandas, Matplotlib and Seaborn
- SQL for Data Analysis
- Python | Math operations for Data analysis
- Python - Data visualization tutorial
- Free Public Data Sets For Analysis

Data Analysis Libraries

- Pandas Tutorial
- NumPy Tutorial - Python Library
- Data Analysis with SciPy

Understanding the Data

- What is Data ?

- Understanding Data Attribute Types | Qualitative and Quantitative
- Univariate, Bivariate and Multivariate data and its analysis
- Attributes and its Types in Data Analytics

Loading the Data

- Pandas Read CSV in Python
- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas

Data Cleaning

- What is Data Cleaning?
- ML | Overview of Data Cleaning
- Best Data Cleaning Techniques for Preparing Your Data

Handling Missing Data

- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- ML | Handling Missing Values
- Working with Missing Data in Pandas
- ML | Handle Missing Data with Simple Imputer
- How to handle missing values of categorical variables in Python?
- Replacing missing values using Pandas in Python

Outliers Detection

- Box Plot
- Detect and Remove the Outliers using Python

- Z score for Outlier Detection - Python
- Clustering-Based approaches for outlier detection in data mining

Exploratory Data Analysis

- What is Exploratory Data Analysis?
- EDA | Exploratory Data Analysis in Python

Time Series Data Analysis

- Time Series Analysis & Visualization in Python
- What is a trend in time series?
- Basic DateTime Operations in Python
- How to deal with missing values in a Timeseries in Python?

Data visualization Tools for Data Analysis

- Tableau Tutorial
- Power BI Tutorial | Learn Power BI

Data Analysis Career Guide

- 30+ Top Data Analytics Projects in 2025 [With Source Codes]
- Top 80+ Data Analyst Interview Questions and Answers
- Data Analyst Roadmap 2025 - A Complete Guide

Data Analysis (Analytics) Tutorial

-
-
-

Data Analysis (Analytics) Tutorial

Data Analysis or Data Analytics is studying, cleaning, modeling, and transforming data to find useful information, suggest conclusions, and support decision-making. This Data Analytics Tutorial will cover

all the basic to advanced concepts of Excel data analysis like data visualization, data preprocessing, time series, data analysis tools, etc.

Data Analysis (Analytics) Tutorial

Data Analysis Tutorial

Data Analysis Process

Data Analysis is developed by the statistician John Tukey in the 1970s. It is a procedure for analyzing data, methods for interpreting the results of such systems, and modes of planning the group of data to make its analysis easier, more accurate, or more factual.

Data Analysis Process

Therefore, data analysis is a process for getting large, unstructured data from different sources and converting it into information that is gone through the below process:

Data Analysis Process

- Data Requirements Specification
- Data Collection
- Data Processing
- Data Cleaning
- Data Analysis
- Communication

Prerequisites for Data Analysis

To strong skill for Data Analysis we needs to learn this resources to have a best practice in this domains.

Prerequisites for Data Analysis

- Python For Data Analysis
- SQL For Data Analysis

- Python Data Visualization
- Data Analysis Datasets

Pandas Tutorial

Learn Pandas to unlock powerful tools for data analysis in Python. This essential library offers versatile data structures like DataFrames, enabling efficient data manipulation, analysis, and visualization. Mastering Pandas will significantly enhance your ability to handle and extract insights from complex datasets, making it an indispensable skill for any data analyst or scientist.

Numpy Tutorial

Learn NumPy to master numerical computing in Python. This foundational library provides support for arrays, matrices, and high-level mathematical functions, making data manipulation and computation highly efficient. Understanding NumPy is crucial for performing advanced data analysis and scientific computing, and it serves as a cornerstone for many other data science libraries.

What is Data?

- Sample Vs Population Statistic
- Different Data Types: Qualitative vs Quantitative (or categorical vs numerical) Univariate Vs Multivariate Data types Attributes of Nominal, Ordinal Vs Interval Data Types
- Qualitative vs Quantitative (or categorical vs numerical)
- Univariate Vs Multivariate Data types
- Attributes of Nominal, Ordinal Vs Interval Data Types

What is Data?

- Qualitative vs Quantitative (or categorical vs numerical)
- Univariate Vs Multivariate Data types
- Attributes of Nominal, Ordinal Vs Interval Data Types

Read and Loading the data set:

- Read Dataset with PandasPython | Read CSVExport Pandas dataframe to a CSV filePandas | Parsing JSON DatasetExporting Pandas DataFrame to JSON FileWorking with Excel files using Pandas
- Python | Read CSV
- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas
- Slicing, Indexing, Manipulating, and Cleaning Pandas Dataframe

Read and Loading the data set:

- Python | Read CSV
- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas

Data Preprocessing:

Data preparation is a critical step in any data analysis or machine learning project. It involves a variety of tasks aimed at transforming raw data into a clean and usable format. Properly prepared data ensures more accurate and reliable analysis results, leading to better decision-making and more effective predictive models. This guide will cover key aspects of data preparation, including data formatting, data cleaning, outlier detection, data transformation, and data sampling.

Data Preprocessing:

- Data FormattingFormatting float column of the data frame in PandasHow to Check the Data Type in Pandas DataFrame?How to change the Pandas datetime format in Python?Convert the column type from string to datetime format in Pandas data frame

- Formatting float column of the data frame in Pandas
- How to Check the Data Type in Pandas DataFrame?
- How to change the Pandas datetime format in Python?
- Convert the column type from string to datetime format in Pandas data frame
- Data Cleaning Overview of Data Cleaning Missing values Working with Missing Data in Pandas Drop rows from Pandas dataframe with missing values or NaN in columns Count NaN or missing values in Pandas DataFrame Handling Missing Values Working with Missing Data Handle Missing Data with Simple Imputer Handle missing values of categorical variables Replacing missing values using Pandas in Python Outliers Detection Boxplots Detect and Remove the Outliers using Python Z-score for outlier Detection Density-based method for outlier Detection Clustering-Based approaches for outlier detection DBSCAN Clustering for outlier detection Binning Binning or Discretization Binning method for data smoothing Isolation Forest for outlier detection Support Vector Machine for outlier detection
- Overview of Data Cleaning
- Missing values Working with Missing Data in Pandas Drop rows from Pandas dataframe with missing values or NaN in columns Count NaN or missing values in Pandas DataFrame Handling Missing Values Working with Missing Data Handle Missing Data with Simple Imputer Handle missing values of categorical variables Replacing missing values using Pandas in Python
- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- Handling Missing Values
- Working with Missing Data
- Handle Missing Data with Simple Imputer
- Handle missing values of categorical variables
- Replacing missing values using Pandas in Python
- Outliers Detection Boxplots Detect and Remove the Outliers using Python Z-score for outlier

Density-based method for outlier Detection
Clustering-Based approaches for outlier detection
DBSCAN Clustering for outlier detection
Binning or Discretization
Binning method for data smoothing
Isolation Forest for outlier detection
Support Vector Machine for outlier detection

- Boxplots

- Detect and Remove the Outliers using Python

- Z-score for outlier Detection

- Density-based method for outlier Detection
Clustering-Based approaches for outlier detection
DBSCAN Clustering for outlier detection

- Clustering-Based approaches for outlier detection

- DBSCAN Clustering for outlier detection

- Binning
Binning or Discretization
Binning method for data smoothing

- Binning or Discretization

- Binning method for data smoothing

- Isolation Forest for outlier detection

- Support Vector Machine for outlier detection

- Data Transformation
Normalization and Scaling
Data Normalization
Difference between Data Normalization and Scaling
Data Normalization with Pandas
How to Standardize Data in a Pandas DataFrame?
Max-Min Normalization
Z-score Normalization
Decimal scaling normalization
Standard Deviation Normalization
Standardization
Log Transformation
Power transformation

- Normalization and Scaling
Data Normalization
Difference between Data Normalization and Scaling
Data Normalization with Pandas
How to Standardize Data in a Pandas DataFrame?
Max-Min Normalization
Z-score Normalization
Decimal scaling normalization
Standard Deviation Normalization
Standardization
Log Transformation
Power transformation

- Data Normalization

- Difference between Data Normalization and Scaling

- Data Normalization with Pandas

- How to Standardize Data in a Pandas DataFrame?

- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation
- Data sampling:Probability samplingSimple Random SamplingClustered SamplingStratified Random samplingSystematic SamplingNon-Probability sampling
- Probability samplingSimple Random SamplingClustered SamplingStratified Random samplingSystematic Sampling
- Simple Random Sampling
- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling
- Non-Probability sampling

Data Preprocessing:

- Formatting float column of the data frame in Pandas
- How to Check the Data Type in Pandas DataFrame?
- How to change the Pandas datetime format in Python?
- Convert the column type from string to datetime format in Pandas data frame

Data Preprocessing:

- Overview of Data Cleaning
- Missing valuesWorking with Missing Data in PandasDrop rows from Pandas dataframe with missing values or NaN in columnsCount NaN or missing values in Pandas DataFrameHandling Missing ValuesWorking with Missing DataHandle Missing Data with Simple ImputerHandle missing

values of categorical variables Replacing missing values using Pandas in Python

- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- Handling Missing Values
- Working with Missing Data
- Handle Missing Data with Simple Imputer
- Handle missing values of categorical variables
- Replacing missing values using Pandas in Python
- Outliers Detection Boxplots Detect and Remove the Outliers using Python Z-score for outlier Detection Density-based method for outlier Detection Clustering-Based approaches for outlier detection DBSCAN Clustering for outlier detection Binning Binning or Discretization Binning method for data smoothing Isolation Forest for outlier detection Support Vector Machine for outlier detection
- Boxplots
- Detect and Remove the Outliers using Python
- Z-score for outlier Detection
- Density-based method for outlier Detection Clustering-Based approaches for outlier detection DBSCAN Clustering for outlier detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Binning Binning or Discretization Binning method for data smoothing
- Binning or Discretization
- Binning method for data smoothing
- Isolation Forest for outlier detection
- Support Vector Machine for outlier detection

Data Preprocessing:

- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- Handling Missing Values
- Working with Missing Data
- Handle Missing Data with Simple Imputer
- Handle missing values of categorical variables
- Replacing missing values using Pandas in Python

Data Preprocessing:

- Boxplots
- Detect and Remove the Outliers using Python
- Z-score for outlier Detection
- Density-based method for outlier Detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Binning
- Binning or Discretization
- Binning method for data smoothing
- Isolation Forest for outlier detection
- Support Vector Machine for outlier detection

Data Preprocessing:

- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection

Data Preprocessing:

- Binning or Discretization
- Binning method for data smoothing

Data Preprocessing:

- Normalization and Scaling
- Difference between Data Normalization and Scaling
- Data Normalization with Pandas
- How to Standardize Data in a Pandas DataFrame?
- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation

Data Preprocessing:

- Data Normalization
- Difference between Data Normalization and Scaling
- Data Normalization with Pandas
- How to Standardize Data in a Pandas DataFrame?
- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization

- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation

Data Preprocessing:

- Probability samplingSimple Random SamplingClustered SamplingStratified Random samplingSystematic Sampling
- Simple Random Sampling
- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling
- Non-Probability sampling

Data Preprocessing:

- Simple Random Sampling
- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is also crucial step in the data analysis process that involves summarizing the main characteristics of a dataset, often with visual methods. The goal of EDA is to understand the data's underlying structure, detect patterns and anomalies, test hypotheses, and check assumptions. EDA is essential for making informed decisions about data preprocessing, feature engineering, and modeling.

Exploratory Data Analysis

- What is Exploratory Data AnalysisExploratory Data Analysis in PythonExploratory Data Analysis in

R

- Exploratory Data Analysis in Python
- Exploratory Data Analysis in R
- Univariate Data EDA: Measures of Central Tendency Measures of spread interquartile range (IQR) ANOVA Skewness Calculate Skewness and Kurtosis in Python Difference between Skewness vs kurtosis Histogram Interpretations of Histogram Boxplots Quantile-Quantile plots Quantile-Quantile Plot in Python Quantile-Quantile Plot in R
- Measures of Central Tendency
- Measures of spread
- interquartile range (IQR)
- ANOVA
- Skewness
- Calculate Skewness and Kurtosis in Python
- Difference between Skewness vs kurtosis
- Histogram
- Interpretations of Histogram
- Boxplots
- Quantile-Quantile plots Quantile-Quantile Plot in Python Quantile-Quantile Plot in R
- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R
- Multivariate Data EDA Cross-tabulation Correlation & Correlation Matrix Correlation and Covariance Factor Analysis Cluster Analysis MANOVA (Multivariate Analysis of Variance) Canonical Correlation Analysis Correspondence Analysis MultiDimensional Scaling
- Cross-tabulation
- Correlation & Correlation Matrix
- Correlation and Covariance
- Factor Analysis

- Cluster Analysis
- MANOVA(Multivariate Analysis of Variance)
- Canonical Correlation Analysis
- Correspondence Analysis
- MultiDimensional Scaling
- Probability DistributionsCentral Limit TheoremCumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in RProbability Density FunctionsProbability Density Estimation & Maximum Likelihood EstimationExponential DistributionNormal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in RBinomial DistributionPython ? Binomial DistributionBernoulli Distribution in RPoisson DistributionP ? ValueZ ? ScoreT-distributionT-distributiont-distribution in Rt-distribution in PythonPoint EstimateConfidence IntervalsChi-Squared TestsHypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test
- Central Limit Theorem
- Cumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in R
- Calculate and plot a Cumulative Distribution function
- Plot Cumulative Distribution Function in R
- Probability Density Functions
- Probability Density Estimation & Maximum Likelihood Estimation
- Exponential Distribution
- Normal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R
- Binomial DistributionPython ? Binomial DistributionBernoulli Distribution in R

- Python ? Binomial Distribution
- Bernoulli Distribution in R
- Poisson Distribution
- P ? Value
- Z ? Score
- T-distributionT-distributiont-distribution in Rt-distribution in Python
- T-distributiont-distribution in Rt-distribution in Python
- t-distribution in R
- t-distribution in Python
- Point Estimate
- Confidence Intervals
- Chi-Squared Tests
- Hypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test
- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Exploratory Data Analysis

- Exploratory Data Analysis in Python
- Exploratory Data Analysis in R

Exploratory Data Analysis

- Measures of Central Tendency
- Measures of spread
- interquartile range (IQR)
- ANOVA
- Skewness
- Calculate Skewness and Kurtosis in Python
- Difference between Skewness vs kurtosis

- Histogram
- Interpretations of Histogram
- Boxplots
- Quantile-Quantile plotsQuantile-Quantile Plot in PythonQuantile-Quantile Plot in R
- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R

Exploratory Data Analysis

- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R

Exploratory Data Analysis

- Cross-tabulation
- Correlation & Correlation Matrix
- Correlation and Covariance
- Factor Analysis
- Cluster Analysis
- MANOVA(Multivariate Analysis of Variance)
- Canonical Correlation Analysis
- Correspondence Analysis
- MultiDimensional Scaling

Exploratory Data Analysis

- Central Limit Theorem
- Cumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in R
- Calculate and plot a Cumulative Distribution function
- Plot Cumulative Distribution Function in R

- Probability Density Functions
- Probability Density Estimation & Maximum Likelihood Estimation
- Exponential Distribution
- Normal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R
- Binomial DistributionPython ? Binomial DistributionBernoulli Distribution in R
- Python ? Binomial Distribution
- Bernoulli Distribution in R
- Poisson Distribution
- P ? Value
- Z ? Score
- T-distributionT-distributiont-distribution in Rt-distribution in Python
- T-distributiont-distribution in Rt-distribution in Python
- t-distribution in R
- t-distribution in Python
- Point Estimate
- Confidence Intervals
- Chi-Squared Tests
- Hypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test
- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Exploratory Data Analysis

- Calculate and plot a Cumulative Distribution function

- Plot Cumulative Distribution Function in R

Exploratory Data Analysis

- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R

Exploratory Data Analysis

- Normal Distribution in Python
- Normal Distribution in R

Exploratory Data Analysis

- Python ? Binomial Distribution
- Bernoulli Distribution in R

Exploratory Data Analysis

- T-distributiont-distribution in Rt-distribution in Python
- t-distribution in R
- t-distribution in Python

Exploratory Data Analysis

- t-distribution in R
- t-distribution in Python

Exploratory Data Analysis

- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Time Series Data Analysis:

Time series data analysis involves examining data points collected or recorded at specific time

intervals. This type of data is ubiquitous in various fields, such as finance, economics, environmental science, and many others. The primary goal is to understand the underlying structure and patterns to make accurate predictions or decisions.

Time Series Data Analysis:

- Define Time Series Data
- Data and Time function in Python
- Time Series Data Plotting
- Deal with missing values in a Time series
- Moving Averages in Time Series Data
- Stationarity in Time Series Data
- Seasonality Detection in Time Series Data
- Trend in Time Series Data
- Testing for Mean Reversion
- Augmented Dickey-Fuller Test
- What is Autocorrelation?

Data Analysis Tools:

- Excel Tutorial
- Tableau Tutorial
- Power BI Tutorial

Need for Data Analysis

Data analytics is significant for business optimization performance. An organization can also use data analytics to make better business decisions and support analyzing customer trends and fulfillment, which can lead to unknown and better products and services. Executing it into the business model indicates businesses can help reduce costs by recognizing more efficient modes of doing business.

Applications of Data Analysis

- Better decision-making: The Key advantage of data analysis is better decision-making in the long term. Rather than depending only on knowledge, businesses are increasingly looking at data before deciding.
- Identification of potential risks: Companies in today's world succeed in high-risk conditions, but those environments require critical risk management processes, and extensive data has contributed to developing new risk management solutions. Data can enhance the effectiveness of actual simulations to predict future risks and create better planning.
- Increase the efficiency of work: Data analysis allows you to analyze a large set of data and present it in a structured way to help reach your organization's objectives. Possibilities and progress within the organization are reflected, and activities can increase work efficiency and productivity. It enables a culture of efficiency and collaboration by allowing managers to share detailed data with employees.
- Delivering relevant products: Products are the oil for every organization, and often the most important asset of organizations. The role of the product management team is to determine trends that drive strategic creation, and activity plans for unique functions and services.
- Track customer behavioral changes: Consumers have a lot to choose from in products available in the markets. Organizations have to pay attention to consumer demands and expectations, So to analyze the behavior of the customer data analysis is very important.

Q.1 What are the four types of Data Analysis?

Answer: There are four types of data Analysis:

Q.1 What are the four types of Data Analysis?

- Descriptive
- Diagnostic
- Predictive
- Prescriptive

Q.2 Why is data analytics so important?

Answer: Data analytics is more than simply showing numbers and figures to the administration. It is about analyzing and understanding your data and using that information to drive actions. Data analytics displays the patterns and trends within the data, which strengthen or otherwise remain unknown.

Q.3 What are the tools useful for data analysis?

Answer: Some of the tools useful for data analysis include:

Q.3 What are the tools useful for data analysis?

- RapidMiner
- KNIME
- Google Search Operators
- Google Fusion Tables
- Solver
- NodeXL
- OpenRefine
- Wolfram Alpha
- io
- Tableau, etc.

Q.4 What are the differences between Data Mining and Data Profiling?

Data Mining | Data Profiling

Data mining is the procedure of finding suitable data that has not yet been determined before. | Data profiling is done to estimate a dataset for its uniqueness, logic, and consistency.

In data mining, raw data is converted into useful information. | It cannot identify incorrect data values.

Data Profiling

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Data Profiting

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Data Profiting

- AI-ML-DS
- Data Analysis
- AI-ML-DS With Python
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge

- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions

- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Prerequisites for Data Analysis

- Exploratory Data Analysis (EDA) with NumPy, Pandas, Matplotlib and Seaborn
- SQL for Data Analysis
- Python | Math operations for Data analysis
- Python - Data visualization tutorial
- Free Public Data Sets For Analysis

Data Analysis Libraries

- Pandas Tutorial
- NumPy Tutorial - Python Library
- Data Analysis with SciPy

Understanding the Data

- What is Data ?
- Understanding Data Attribute Types | Qualitative and Quantitative
- Univariate, Bivariate and Multivariate data and its analysis
- Attributes and its Types in Data Analytics

Loading the Data

- Pandas Read CSV in Python
- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas

Data Cleaning

- What is Data Cleaning?

- ML | Overview of Data Cleaning
- Best Data Cleaning Techniques for Preparing Your Data

Handling Missing Data

- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- ML | Handling Missing Values
- Working with Missing Data in Pandas
- ML | Handle Missing Data with Simple Imputer
- How to handle missing values of categorical variables in Python?
- Replacing missing values using Pandas in Python

Outliers Detection

- Box Plot
- Detect and Remove the Outliers using Python
- Z score for Outlier Detection - Python
- Clustering-Based approaches for outlier detection in data mining

Exploratory Data Analysis

- What is Exploratory Data Analysis?
- EDA | Exploratory Data Analysis in Python

Time Series Data Analysis

- Time Series Analysis & Visualization in Python
- What is a trend in time series?
- Basic DateTime Operations in Python
- How to deal with missing values in a Timeseries in Python?

Data visualization Tools for Data Analysis

- Tableau Tutorial
- Power BI Tutorial | Learn Power BI

Data Analysis Career Guide

- 30+ Top Data Analytics Projects in 2025 [With Source Codes]
- Top 80+ Data Analyst Interview Questions and Answers
- Data Analyst Roadmap 2025 - A Complete Guide

Data Analysis (Analytics) Tutorial

-
-
-

Data Analysis (Analytics) Tutorial

Data Analysis or Data Analytics is studying, cleaning, modeling, and transforming data to find useful information, suggest conclusions, and support decision-making. This Data Analytics Tutorial will cover all the basic to advanced concepts of Excel data analysis like data visualization, data preprocessing, time series, data analysis tools, etc.

Data Analysis (Analytics) Tutorial

Data Analysis Tutorial

Data Analysis Process

Data Analysis is developed by the statistician John Tukey in the 1970s. It is a procedure for analyzing data, methods for interpreting the results of such systems, and modes of planning the group of data to make its analysis easier, more accurate, or more factual.

Data Analysis Process

Therefore, data analysis is a process for getting large, unstructured data from different sources and

converting it into information that is gone through the below process:

Data Analysis Process

- Data Requirements Specification
- Data Collection
- Data Processing
- Data Cleaning
- Data Analysis
- Communication

Prerequisites for Data Analysis

To strong skill for Data Analysis we needs to learn this resources to have a best practice in this domains.

Prerequisites for Data Analysis

- Python For Data Analysis
- SQL For Data Analysis
- Python Data Visulization
- Data Analysis Datasets

Pandas Tutorial

Learn Pandas to unlock powerful tools for data analysis in Python. This essential library offers versatile data structures like DataFrames, enabling efficient data manipulation, analysis, and visualization. Mastering Pandas will significantly enhance your ability to handle and extract insights from complex datasets, making it an indispensable skill for any data analyst or scientist.

Numpy Tutorial

Learn NumPy to master numerical computing in Python. This foundational library provides support for arrays, matrices, and high-level mathematical functions, making data manipulation and

computation highly efficient. Understanding NumPy is crucial for performing advanced data analysis and scientific computing, and it serves as a cornerstone for many other data science libraries.

What is Data?

- Sample Vs Population Statistic
- Different Data Types: Qualitative vs Quantitative (or categorical vs numerical) Univariate Vs Multivariate Data types Attributes of Nominal, Ordinal Vs Interval Data Types
- Qualitative vs Quantitative (or categorical vs numerical)
- Univariate Vs Multivariate Data types
- Attributes of Nominal, Ordinal Vs Interval Data Types

What is Data?

- Qualitative vs Quantitative (or categorical vs numerical)
- Univariate Vs Multivariate Data types
- Attributes of Nominal, Ordinal Vs Interval Data Types

Read and Loading the data set:

- Read Dataset with Pandas Python | Read CSV Export Pandas dataframe to a CSV file Pandas | Parsing JSON Dataset Exporting Pandas DataFrame to JSON File Working with Excel files using Pandas
- Python | Read CSV
- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas
- Slicing, Indexing, Manipulating, and Cleaning Pandas Dataframe

Read and Loading the data set:

- Python | Read CSV

- Export Pandas dataframe to a CSV file
- Pandas | Parsing JSON Dataset
- Exporting Pandas DataFrame to JSON File
- Working with Excel files using Pandas

Data Preprocessing:

Data preparation is a critical step in any data analysis or machine learning project. It involves a variety of tasks aimed at transforming raw data into a clean and usable format. Properly prepared data ensures more accurate and reliable analysis results, leading to better decision-making and more effective predictive models. This guide will cover key aspects of data preparation, including data formatting, data cleaning, outlier detection, data transformation, and data sampling.

Data Preprocessing:

- Data Formatting
Formatting float column of the data frame in Pandas
How to Check the Data Type in Pandas DataFrame?
How to change the Pandas datetime format in Python?
Convert the column type from string to datetime format in Pandas data frame
- Formatting float column of the data frame in Pandas
- How to Check the Data Type in Pandas DataFrame?
- How to change the Pandas datetime format in Python?
- Convert the column type from string to datetime format in Pandas data frame
- Data Cleaning
Overview of Data Cleaning
Missing values
Working with Missing Data in Pandas
Drop rows from Pandas dataframe with missing values or NaN in columns
Count NaN or missing values in Pandas DataFrame
Handling Missing Values
Working with Missing Data
Handle Missing Data with Simple Imputer
Handle missing values of categorical variables
Replacing missing values using Pandas in Python
Outliers Detection
Boxplots
Detect and Remove the Outliers using Python
Z-score for outlier Detection
Density-based method for outlier Detection
Clustering-Based approaches for outlier detection
DBSCAN Clustering for outlier detection
Binning
Binning or Discretization
Binning method for data smoothing
Isolation Forest for outlier detection
Support Vector Machine for outlier

detection

- Overview of Data Cleaning

- Missing valuesWorking with Missing Data in PandasDrop rows from Pandas dataframe with missing values or NaN in columnsCount NaN or missing values in Pandas DataFrameHandling Missing ValuesWorking with Missing DataHandle Missing Data with Simple ImputerHandle missing values of categorical variablesReplacing missing values using Pandas in Python

- Working with Missing Data in Pandas

- Drop rows from Pandas dataframe with missing values or NaN in columns

- Count NaN or missing values in Pandas DataFrame

- Handling Missing Values

- Working with Missing Data

- Handle Missing Data with Simple Imputer

- Handle missing values of categorical variables

- Replacing missing values using Pandas in Python

- Outliers DetectionBoxplotsDetect and Remove the Outliers using PythonZ-score for outlier DetectionDensity-based method for outlier DetectionClustering-Based approaches for outlier detectionDBSCAN Clustering for outlier detectionBinningBinning or DiscretizationBinning method for data smoothingIsolation Forest for outlier detectionSupport Vector Machine for outlier detection

- Boxplots

- Detect and Remove the Outliers using Python

- Z-score for outlier Detection

- Density-based method for outlier DetectionClustering-Based approaches for outlier detectionDBSCAN Clustering for outlier detection

- Clustering-Based approaches for outlier detection

- DBSCAN Clustering for outlier detection

- BinningBinning or DiscretizationBinning method for data smoothing

- Binning or Discretization

- Binning method for data smoothing
- Isolation Forest for outlier detection
- Support Vector Machine for outlier detection
- Data Transformation
 - Normalization and Scaling
 - Data Normalization
 - Difference between Data Normalization and Scaling
 - Data Normalization with Pandas
 - How to Standardize Data in a Pandas DataFrame?
 - Max-Min Normalization
 - Z-score Normalization
 - Decimal scaling normalization
 - Standard Deviation Normalization
 - Standardization
 - Log Transformation
 - Power transformation
- Data Normalization
- Difference between Data Normalization and Scaling
- Data Normalization with Pandas
- How to Standardize Data in a Pandas DataFrame?
- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation
- Data sampling:
 - Probability sampling
 - Simple Random Sampling
 - Clustered Sampling
 - Stratified Random sampling
 - Systematic Sampling
 - Non-Probability sampling
- Probability sampling
 - Simple Random Sampling
 - Clustered Sampling
 - Stratified Random sampling
 - Systematic Sampling
- Simple Random Sampling

- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling
- Non-Probability sampling

Data Preprocessing:

- Formatting float column of the data frame in Pandas
- How to Check the Data Type in Pandas DataFrame?
- How to change the Pandas datetime format in Python?
- Convert the column type from string to datetime format in Pandas data frame

Data Preprocessing:

- Overview of Data Cleaning
- Missing valuesWorking with Missing Data in PandasDrop rows from Pandas dataframe with missing values or NaN in columnsCount NaN or missing values in Pandas DataFrameHandling Missing ValuesWorking with Missing DataHandle Missing Data with Simple ImputerHandle missing values of categorical variablesReplacing missing values using Pandas in Python
- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- Handling Missing Values
- Working with Missing Data
- Handle Missing Data with Simple Imputer
- Handle missing values of categorical variables
- Replacing missing values using Pandas in Python
- Outliers DetectionBoxplotsDetect and Remove the Outliers using PythonZ-score for outlier DetectionDensity-based method for outlier DetectionClustering-Based approaches for outlier detectionDBSCAN Clustering for outlier detectionBinningBinning or DiscretizationBinning method for

data smoothing Isolation Forest for outlier detection Support Vector Machine for outlier detection

- Boxplots
- Detect and Remove the Outliers using Python
- Z-score for outlier Detection
- Density-based method for outlier Detection Clustering-Based approaches for outlier detection DBSCAN Clustering for outlier detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Binning Binning or Discretization Binning method for data smoothing
- Binning or Discretization
- Binning method for data smoothing
- Isolation Forest for outlier detection
- Support Vector Machine for outlier detection

Data Preprocessing:

- Working with Missing Data in Pandas
- Drop rows from Pandas dataframe with missing values or NaN in columns
- Count NaN or missing values in Pandas DataFrame
- Handling Missing Values
- Working with Missing Data
- Handle Missing Data with Simple Imputer
- Handle missing values of categorical variables
- Replacing missing values using Pandas in Python

Data Preprocessing:

- Boxplots
- Detect and Remove the Outliers using Python
- Z-score for outlier Detection

- Density-based method for outlier Detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection
- Binning
- Binning or Discretization
- Binning method for data smoothing
- Isolation Forest for outlier detection
- Support Vector Machine for outlier detection

Data Preprocessing:

- Clustering-Based approaches for outlier detection
- DBSCAN Clustering for outlier detection

Data Preprocessing:

- Binning or Discretization
- Binning method for data smoothing

Data Preprocessing:

- Normalization and Scaling
- Data Normalization
- Difference between Data Normalization and Scaling
- Data Normalization with Pandas
- How to Standardize Data in a Pandas DataFrame?
- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation

- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation

Data Preprocessing:

- Data Normalization
- Difference between Data Normalization and Scaling
- Data Normalization with Pandas
- How to Standardize Data in a Pandas DataFrame?
- Max-Min Normalization
- Z-score Normalization
- Decimal scaling normalization
- Standard Deviation Normalization
- Standardization
- Log Transformation
- Power transformation

Data Preprocessing:

- Probability samplingSimple Random SamplingClustered SamplingStratified Random samplingSystematic Sampling
- Simple Random Sampling
- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling
- Non-Probability sampling

Data Preprocessing:

- Simple Random Sampling
- Clustered Sampling
- Stratified Random sampling
- Systematic Sampling

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is also crucial step in the data analysis process that involves summarizing the main characteristics of a dataset, often with visual methods. The goal of EDA is to understand the data's underlying structure, detect patterns and anomalies, test hypotheses, and check assumptions. EDA is essential for making informed decisions about data preprocessing, feature engineering, and modeling.

Exploratory Data Analysis

- What is Exploratory Data Analysis
- Exploratory Data Analysis in Python
- Exploratory Data Analysis in R
- Exploratory Data Analysis in Python
- Exploratory Data Analysis in R
- Univariate Data EDA: Measures of Central Tendency Measures of spread interquartile range (IQR) ANOVA Skewness Calculate Skewness and Kurtosis in Python Difference between Skewness vs kurtosis Histogram Interpretations of Histogram Boxplots Quantile-Quantile plots Quantile-Quantile Plot in Python Quantile-Quantile Plot in R
- Measures of Central Tendency
- Measures of spread
- interquartile range (IQR)
- ANOVA
- Skewness
- Calculate Skewness and Kurtosis in Python

- Difference between Skewness vs kurtosis
- Histogram
- Interpretations of Histogram
- Boxplots
- Quantile-Quantile plotsQuantile-Quantile Plot in PythonQuantile-Quantile Plot in R
- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R
- Multivariate Data EDACross-tabulationCorrelation & Correlation MatrixCorrelation and CovarianceFactor AnalysisCluster AnalysisMANOVA(Multivariate Analysis of Variance)Canonical Correlation AnalysisCorrespondence AnalysisMultiDimensional Scaling
- Cross-tabulation
- Correlation & Correlation Matrix
- Correlation and Covariance
- Factor Analysis
- Cluster Analysis
- MANOVA(Multivariate Analysis of Variance)
- Canonical Correlation Analysis
- Correspondence Analysis
- MultiDimensional Scaling
- Probability DistributionsCentral Limit TheoremCumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in RProbability Density FunctionsProbability Density Estimation & Maximum Likelihood EstimationExponential DistributionNormal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in RBinomial DistributionPython ? Binomial DistributionBernoulli Distribution in RPoisson DistributionP ? ValueZ ? ScoreT-distributionT-distributiont-distribution in Rt-distribution in PythonPoint EstimateConfidence IntervalsChi-Squared TestsHypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test

- Central Limit Theorem
- Cumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in R
- Calculate and plot a Cumulative Distribution function
- Plot Cumulative Distribution Function in R
- Probability Density Functions
- Probability Density Estimation & Maximum Likelihood Estimation
- Exponential Distribution
- Normal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R
- Binomial DistributionPython ? Binomial DistributionBernoulli Distribution in R
- Python ? Binomial Distribution
- Bernoulli Distribution in R
- Poisson Distribution
- P ? Value
- Z ? Score
- T-distributionT-distributiont-distribution in Rt-distribution in Python
- T-distributiont-distribution in Rt-distribution in Python
- t-distribution in R
- t-distribution in Python
- Point Estimate
- Confidence Intervals
- Chi-Squared Tests
- Hypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test

- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Exploratory Data Analysis

- Exploratory Data Analysis in Python
- Exploratory Data Analysis in R

Exploratory Data Analysis

- Measures of Central Tendency
- Measures of spread
- interquartile range (IQR)
- ANOVA
- Skewness
- Calculate Skewness and Kurtosis in Python
- Difference between Skewness vs kurtosis
- Histogram
- Interpretations of Histogram
- Boxplots
- Quantile-Quantile plotsQuantile-Quantile Plot in PythonQuantile-Quantile Plot in R
- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R

Exploratory Data Analysis

- Quantile-Quantile Plot in Python
- Quantile-Quantile Plot in R

Exploratory Data Analysis

- Cross-tabulation
- Correlation & Correlation Matrix

- Correlation and Covariance
- Factor Analysis
- Cluster Analysis
- MANOVA(Multivariate Analysis of Variance)
- Canonical Correlation Analysis
- Correspondence Analysis
- MultiDimensional Scaling

Exploratory Data Analysis

- Central Limit Theorem
- Cumulative Distribution FunctionsCalculate and plot a Cumulative Distribution functionPlot Cumulative Distribution Function in R
- Calculate and plot a Cumulative Distribution function
- Plot Cumulative Distribution Function in R
- Probability Density Functions
- Probability Density Estimation & Maximum Likelihood Estimation
- Exponential Distribution
- Normal DistributionNormal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R
- Binomial DistributionPython ? Binomial DistributionBernoulli Distribution in R
- Python ? Binomial Distribution
- Bernoulli Distribution in R
- Poisson Distribution
- P ? Value

- Z ? Score
- T-distributionT-distributiont-distribution in Rt-distribution in Python
- T-distributiont-distribution in Rt-distribution in Python
- t-distribution in R
- t-distribution in Python
- Point Estimate
- Confidence Intervals
- Chi-Squared Tests
- Hypothesis TestingHypothesis Testing FormulaOne-Tailed Test & Two-Tailed Test
- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Exploratory Data Analysis

- Calculate and plot a Cumulative Distribution function
- Plot Cumulative Distribution Function in R

Exploratory Data Analysis

- Normal Distribution functionNormal Distribution in PythonNormal Distribution in R
- Normal Distribution in Python
- Normal Distribution in R

Exploratory Data Analysis

- Normal Distribution in Python
- Normal Distribution in R

Exploratory Data Analysis

- Python ? Binomial Distribution
- Bernoulli Distribution in R

Exploratory Data Analysis

- T-distribution
- t-distribution in R
- t-distribution in Python

Exploratory Data Analysis

- t-distribution in R
- t-distribution in Python

Exploratory Data Analysis

- Hypothesis Testing Formula
- One-Tailed Test & Two-Tailed Test

Time Series Data Analysis:

Time series data analysis involves examining data points collected or recorded at specific time intervals. This type of data is ubiquitous in various fields, such as finance, economics, environmental science, and many others. The primary goal is to understand the underlying structure and patterns to make accurate predictions or decisions.

Time Series Data Analysis:

- Define Time Series Data
- Data and Time function in Python
- Time Series Data Plotting
- Deal with missing values in a Time series
- Moving Averages in Time Series Data
- Stationarity in Time Series Data
- Seasonality Detection in Time Series Data
- Trend in Time Series Data
- Testing for Mean Reversion

- Augmented Dickey-Fuller Test
- What is Autocorrelation?

Data Analysis Tools:

- Excel Tutorial
- Tableau Tutorial
- Power BI Tutorial

Need for Data Analysis

Data analytics is significant for business optimization performance. An organization can also use data analytics to make better business decisions and support analyzing customer trends and fulfillment, which can lead to unknown and better products and services. Executing it into the business model indicates businesses can help reduce costs by recognizing more efficient modes of doing business.

Applications of Data Analysis

- Better decision-making: The Key advantage of data analysis is better decision-making in the long term. Rather than depending only on knowledge, businesses are increasingly looking at data before deciding.
- Identification of potential risks: Companies in today's world succeed in high-risk conditions, but those environments require critical risk management processes, and extensive data has contributed to developing new risk management solutions. Data can enhance the effectiveness of actual simulations to predict future risks and create better planning.
- Increase the efficiency of work: Data analysis allows you to analyze a large set of data and present it in a structured way to help reach your organization's objectives. Possibilities and progress within the organization are reflected, and activities can increase work efficiency and productivity. It enables a culture of efficiency and collaboration by allowing managers to share detailed data with employees.

- Delivering relevant products:Products are the oil for every organization, and often the most important asset of organizations. The role of the product management team is to determine trends that drive strategic creation, and activity plans for unique functions and services.
- Track customer behavioral changes:Consumers have a lot to choose from in products available in the markets. Organizations have to pay attention to consumer demands and expectations, So to analyze the behavior of the customer data analysis is very important.

Q.1 What are the four types of Data Analysis?

Answer:There are four types of data Analysis:

Q.1 What are the four types of Data Analysis?

- Descriptive
- Diagnostic
- Predictive
- Prescriptive

Q.2 Why is data analytics so important?

Answer:Data analytics is more than simply showing numbers and figures to the administration. It is about analyzing and understanding your data and using that information to drive actions. Data analytics displays the patterns and trends within the data, which strengthen or otherwise remain unknown.

Q.3 What are the tools useful for data analysis?

Answer:Some of the tools useful for data analysis include:

Q.3 What are the tools useful for data analysis?

- RapidMiner
- KNIME
- Google Search Operators

- Google Fusion Tables
- Solver
- NodeXL
- OpenRefine
- Wolfram Alpha
- io
- Tableau, etc.

Q.4 What are the differences between Data Mining and Data Profiling?

Data Mining | Data Profiting

Data mining is the procedure of finding suitable data that has not yet been determined before. | Data profiling is done to estimate a dataset for its uniqueness, logic, and consistency.

In data mining, raw data is converted into useful information. | It cannot identify incorrect data values.

Data Profiting

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Data Profiting

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Data Profiting

- AI-ML-DS
- Data Analysis
- AI-ML-DS With Python
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python

- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP

- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems

- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner

- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming

- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is Data Engineering?

-
-
-

What is Data Engineering?

Data engineering forms the backbone of modern data-driven enterprises, encompassing the design, development, and maintenance of crucial systems and infrastructure for managing data throughout its lifecycle.

What is Data Engineering?

In this article, we will explore key aspects of data engineering, its key features, importance, and the distinctions between data engineering and data science.

What is Data Engineering?

What is Data Engineering?

Table of Content

What is Data Engineering?

- What Is Data Engineering?
- Key Components of Data Engineering
 - 1. Data Collection
 - 2. Data Storage
 - 3. Data Processing
 - 4. Data Pipelines
 - 5. Data Quality and Governance
- Why Is Data Engineering Important?
- Core Responsibilities of a Data Engineer
- Why Does Data Need Processing through Data Engineering?
- Tools and Technologies in Data Engineering
- Challenges in Data Engineering
- Data Engineering vs. Data Science
- Future Trends in Data Engineering
- Conclusion
- FAQs on Data Engineering

What is Data Engineering?

- 1. Data Collection

- 2. Data Storage
- 3. Data Processing
- 4. Data Pipelines
- 5. Data Quality and Governance

What Is Data Engineering?

The field of data engineering is concerned with designing, constructing, and maintaining the support systems and infrastructures necessary for data inbound, storage, processing, and analysis. Data engineers manage huge sets of data in a real-time environment. They are supposed to provide high quality of information that is really usable by different business departments.

What Is Data Engineering?

Data engineers deal with large volumes of data, often in real-time, and their role is crucial in enabling businesses to extract valuable insights from their data assets. They work closely with data scientists, analysts, and other stakeholders to ensure that the data infrastructure supports the organization's goals and requirements.

1. Data Collection

Data engineering starts with data collection, which involves gathering raw data from various sources such as databases, APIs, sensors, and logs. This step is crucial as the quality and completeness of collected data directly impact subsequent processes.

2. Data Storage

Once data is collected, it needs to be stored in a manner that allows for efficient retrieval and processing. Data engineers design and manage storage solutions such as data warehouses, data lakes, and databases. These solutions must balance performance, scalability, and cost-effectiveness.

3. Data Processing

Data processing involves transforming raw data into a structured and usable format. This includes data cleaning, normalization, and integration. Data engineers use tools like Apache Spark, Hadoop, and ETL (Extract, Transform, Load) frameworks to automate and optimize these processes.

4. Data Pipelines

Data pipelines are automated workflows that move data from source to destination, ensuring that data flows smoothly and consistently. They encompass data extraction, transformation, and loading (ETL), as well as real-time data streaming. Effective data pipeline management is essential for maintaining data integrity and availability.

5. Data Quality and Governance

Ensuring data quality and governance involves implementing policies and procedures to maintain data accuracy, consistency, and security. Data engineers establish data validation checks, monitor data for anomalies, and enforce compliance with data privacy regulations.

Why Is Data Engineering Important?

Data engineering forms the backbone of any data-driven enterprise. It ensures that data is accurate, reliable, and accessible, providing a solid foundation for data analysis, machine learning, and artificial intelligence applications. Without effective data engineering, organizations may struggle with data inconsistencies, bottlenecks, and inefficiencies, hindering their ability to derive meaningful insights.

Core Responsibilities of a Data Engineer

Data engineers perform many information engineering duties such as managing data from the cradle to the grave. Here are some key responsibilities of data engineers:

Core Responsibilities of a Data Engineer

Here are some key responsibilities of data engineers:

Core Responsibilities of a Data Engineer

- **Data Collection:**With this, they provide the platform to designers and developers to draw data from database, applications, APIs, external feedbacks, and sources. These data will then be processed using a series of algorithms.
- **Data Storage:**Depending on their preference, data engineers opt for the most suitable data storage facilities like databases (SQL, NoSQL), data lakes and warehouses for the safe and proper storage of the collected data.
- **Data Processing:**They set up and maintain data pipelines as well as ETL processes to the end of clean up, transformation, and preprocessing of raw data in a manner that enables the data to be analyzed and reported.
- **Data Integration:**Engineers of data combine data channels within one system, getting the holistic and verified data stream.
- **Data Quality and Governance:**They check data quality, data validation rules, and monitoring mechanisms. This is performed so as to know data integrity problems, discover anomalies and if any data quality issues may be present. They build tools to control data quality, integrity, and security that is done through data validation, error handling and compliance with regulations like general data protection, HIPAA, among others.
- **Performance Optimization:**Data engineers bring about the most efficient ways of running data processing workflows, queries, and database performance to aim at the fastest, most efficient and scalable data operations.

Why Does Data Need Processing through Data Engineering?

Data requires processing through data engineering to transform it from its raw, often disparate form into a structured and usable format for analysis and decision-making. In its raw state, data may be fragmented, inconsistent, and laden with errors, rendering it unsuitable for meaningful insights. Data engineering plays a pivotal role in rectifying these shortcomings by employing a series of processes aimed at cleansing, integrating, and enhancing the data. By ensuring data quality, consistency, and accessibility, data engineering lays the groundwork for effective analytics, enabling organizations to

extract valuable insights, optimize operations, and drive informed decision-making. In essence, data processing through data engineering acts as the gateway to unlocking the full potential of data assets within an organization.

Why Does Data Need Processing through Data Engineering?

About processing of data through data engineering this is not only so for a few key reasons but also important for several of them.

Why Does Data Need Processing through Data Engineering?

- **Data Quality Improvement:**Raw data has its own errors, gaps, and inconsistency issues. Data engineering processes, e.g., data cleaning, normalization, and validation provide solutions to the issues by means of locating the issues and correcting them, thereby making data accurate, complete and reliable.
- **Scalability and Performance:**Data engineering builds high-capacity data pipelines and processing algorithms that can tackle the challenge of huge data volumes effectively. Data engineering which normally refers to the optimizing of the data processing and storage systems helps to streamline data operations to the point where it can be processed timely and be used in the decision-making process and real-time analytics.
- **Data Governance and Compliance:**Data engineering ensures the development of comprehensible, transparent, coherent, and consistent data governance policies, security measures and requirements according to GDPR, HIPAA, and industry standards. This means that the necessary measures should be applied such as data privacy, confidentiality, and integrity. Also the access control and audit trails on the changes to be made on the data usage should be implemented.
- **Support for Data Science and Analytics:**Data engineering as such would be concerned with preparation and pre-processing of data for professionals in data science and analysis areas thus providing them with clean and tailored datasets for advanced analytics, ML, time-series and AI applications. It thereby makes possible data mining and provides organizations the ability to get information that is actionable based on data.

Tools and Technologies in Data Engineering

Data engineers leverage a variety of tools and technologies to build and maintain data infrastructure. Some of the key tools include:

Tools and Technologies in Data Engineering

- Database Management Systems (DBMS):MySQL, PostgreSQL, MongoDB
- Data Warehousing Solutions:Amazon Redshift, Google BigQuery, Snowflake
- Big Data Technologies:Apache Hadoop, Apache Spark
- ETL Tools:Talend, Apache Nifi, Microsoft Azure Data Factory
- Data Orchestration Tools:Apache Airflow, Prefect, Luigi

Challenges in Data Engineering

Data engineering is not without its challenges. Common issues include:

Challenges in Data Engineering

- Handling Large Volumes of Data:Managing and processing large datasets efficiently requires specialized tools and techniques.
- Ensuring Data Quality:Consistently maintaining high data quality across diverse sources can be complex.
- Scalability:Building systems that can scale with growing data volumes and user demands.
- Data Security:Protecting sensitive data from breaches and ensuring compliance with regulations.
- Keeping Up with Technology:Rapid advancements in technology require data engineers to continually update their skills and knowledge.

Data Engineering vs. Data Science

Data engineering and data science are two distinct but closely related disciplines within the field of data analytics.

Data Engineering vs. Data Science

Aspect | Data Engineering | Data Science

Focus | Data infrastructure, pipelines, and processing | Data analysis, modeling, and insights

Objective | Prepare, transform, and manage data for use | Extract insights, build predictive models

Data Handling | Raw data cleaning, integration, storage | Analyzing, exploring, visualizing data

Tools and Technologies | Apache Hadoop, Spark, Kafka, SQL/NoSQL databases | Python/R, Jupyter Notebooks, Machine Learning libraries

Skills | Programming (Python, Java), ETL, database management | Statistics, Machine Learning, Data Visualization

Output | Clean, structured data ready for analysis and reporting | Predictive models, insights, actionable recommendations

Role | Develop and maintain data pipelines, ensure data quality | Analyze data, build ML models, communicate findings

Use Cases | Data integration, ETL processes, data warehousing | Predictive analytics, recommendation systems

Future Trends in Data Engineering

The field of data engineering is continually evolving. Some emerging trends include:

Future Trends in Data Engineering

- DataOps: An extension of DevOps, focusing on improving collaboration and automation in data workflows.
- Real-Time Data Processing: Increasing demand for real-time analytics and decision-making capabilities.
- Machine Learning Operations (MLOps): Integrating machine learning models into data pipelines for seamless deployment and management.
- Cloud-Native Data Engineering: Leveraging cloud platforms for scalable and cost-effective data solutions.
- Data Privacy and Ethics: Growing emphasis on data privacy, ethical data usage, and compliance

with regulations like GDPR and CCPA.

Conclusion

In conclusion, data Engineering is the basis of current data-driven enterprises that are managing data infrastructure and processes including design, development, and running. It aids the gather, stock, treatment, and interlinking of vast quantities of data from different resources, this gives rise to their availability, accuracy, and reliability suitable for analysis and decision making.

What is the difference between data engineering and data science?

The main task of data engineering is to design and construct data infrastructure and engineering systems, development, aggregation, and implementation of pipelines and processes to collect, store, and prepare data to be analyzed. The data science is to the contrary, refers to analyzing data, modeling, making prediction, and data mining aiming to provide insights to inform decision-making and business objectives.

What skills are required for a career in data engineering?

The necessary skillset of data engineers encompass a good grasp of programming languages (Python, Java, Scala), databases (SQL, NoSQL), comprehension of big data technologies (Hadoop, Spark), data modeling, ETL processes, cloud computing , and the problem-solving abilities,.

What tools are commonly used in data engineering?

Some of typical tools in data engineering are Apache Hadoop, Apache Spark, Apache Kafka, SQL/NoSQL databases (e.g., PostgreSQL, MongoDB) ,ETL tools (e.g., Talend, Informatica), cloud platforms (AWS, Azure, GCP) and data pipeline orchestration tools (e.g., Apache Airflow).

How does data engineering contribute to data-driven decision-making?

Data engineering guarantees that data to be used for analysis is easy to access, correct, and consistent, hence giving a chance to the data scientists and an analysts to develop insights and build models or choose data-driven decisions. This consists in designing scalable data system,

enhancing data processing workflows and holding data integrity and governance issues.

How does data engineering contribute to data-driven decision-making?

Fast-Track Your Data Engineering Career! Learn how to design data pipelines and work with tools for big data. This course will equip you with the expertise to tackle real-world data challenges. Enroll today and start your journey in data engineering!

How does data engineering contribute to data-driven decision-making?

A

How does data engineering contribute to data-driven decision-making?

- Data Engineering
- AI-ML-DS

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge

- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions

- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to PySpark | Distributed Computing with Apache Spark

-
-
-

Introduction to PySpark | Distributed Computing with Apache Spark

Datasets are becoming huge. Infact, data is growing faster than processing speeds. Therefore, algorithms involving large data and high amount of computation are often run on a distributed computing system. A distributed computing system involves nodes (networked computers) that run processes in parallel and communicate (if, necessary).

Introduction to PySpark | Distributed Computing with Apache Spark

MapReduce? The programming model that is used for Distributed computing is known as MapReduce. The MapReduce model involves two stages, Map and Reduce.

Introduction to PySpark | Distributed Computing with Apache Spark

- Map? The mapper processes each line of the input data (it is in the form of a file), and produces key ? value pairs. Input data ? Mapper ? list([key, value])
- Reduce? The reducer processes the list of key ? value pairs (after the Mapper?s function). It outputs a new set of key ? value pairs. list([key, value]) ? Reducer ? list([key, list(values)])

Introduction to PySpark | Distributed Computing with Apache Spark

Spark? Spark (open source Big-Data processing engine by Apache) is a cluster computing system. It is faster as compared to other cluster computing systems (such as, Hadoop). It provides high level APIs in Python, Scala, and Java. Parallel jobs are easy to write in Spark. We will cover PySpark (Python + Apache Spark), because this will make the learning curve flatter. To install Spark on a linux system, follow[this](#). To run Spark in a multi ? cluster system, follow[this](#). We will see how to

create RDDs (fundamental data structure of Spark).

Introduction to PySpark | Distributed Computing with Apache Spark

RDDs (Resilient Distributed Datasets)? RDDs are immutable collection of objects. Since we are using PySpark, these objects can be of multiple types. These will become more clear further.

Introduction to PySpark | Distributed Computing with Apache Spark

SparkContext? For creating a standalone application in Spark, we first define a SparkContext ?

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

```
frompysparkimportSparkConf,
```

```
SparkContextconf=SparkConf().setMaster("local").setAppName("Test")# setMaster(local) - we are  
doing tasks on a single machinesc=SparkContext(conf=conf)
```

Introduction to PySpark | Distributed Computing with Apache Spark

RDD transformations? Now, a SparkContext object is created. Now, we will create RDDs and see some transformations on them.

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

```
# create an RDD called lines from ?file_name.txt?lines=sc.textFile("file_name.txt",2)# print
```

lines.collect() prints the whole RDD
lines.collect()

Introduction to PySpark | Distributed Computing with Apache Spark

One major advantage of using Spark is that it does not load the dataset into memory, lines is a pointer to the file_name.txt file.

Introduction to PySpark | Distributed Computing with Apache Spark

A simple PySpark app to count the degree of each vertex for a given graph?

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

Introduction to PySpark | Distributed Computing with Apache Spark

```
from pyspark import SparkConf,
```

```
SparkContext conf=SparkConf().setMaster("local").setAppName("Test")# setMaster(local) - we are  
doing tasks on a single
```

```
machines sc=SparkContext(conf=conf) def conv(line): line=line.split() return (int(line[0]),
```

```
[int(line[1])]) def numNeighbours(x,
```

```
y): return len(x)+len(y) lines=sc.textFile('graph.txt') edges=lines.map(lambda line:
```

```
conv(line)) Adj_list=edges.reduceByKey(lambda x, y: numNeighbours(x, y)) print Adj_list.collect()
```

Introduction to PySpark | Distributed Computing with Apache Spark

Understanding the above code?

Introduction to PySpark | Distributed Computing with Apache Spark

- Our text file is in the following format ? (each line represents an edge of a directed graph)
1 21 32 33 4.PySpark

- Large Datasets may contain millions of nodes, and edges.

- First few lines set up the SparkContext. We create an RDDlinesfrom it.
- Then, we transform thelinesRDD toedgesRDD.The function conv acts on each line and key value pairs of the form (1, 2), (1, 3), (2, 3), (3, 4), ? are stored in theedgesRDD.
- After this thereduceByKeyaggregates all the key ? pairs corresponding to a particular key andnumNeighboursfunction is used for generating each vertex?s degree in a separate RDDAdj_list, which has the form (1, 2), (2, 1), (3, 1), ?

Introduction to PySpark | Distributed Computing with Apache Spark

Running the code?

Introduction to PySpark | Distributed Computing with Apache Spark

- The above code can be run by the following commands ?\$ cd /home/arik/Downloads/spark-1.6.0/
\$./bin/spark-submit degree.py
- You can use your Spark installation path in the first line.

Introduction to PySpark | Distributed Computing with Apache Spark

We will see more on, how to run MapReduce tasks in a cluster of machines using Spark, and also go through other MapReduce tasks.

Introduction to PySpark | Distributed Computing with Apache Spark

References?

Introduction to PySpark | Distributed Computing with Apache Spark

- <http://lntool.github.io/SparkTutorial/>
- <https://spark.apache.org/>

Introduction to PySpark | Distributed Computing with Apache Spark

Become a Full-Stack Developer and also get 90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

Introduction to PySpark | Distributed Computing with Apache Spark

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Introduction to PySpark | Distributed Computing with Apache Spark

- Data Science
- Misc
- Web Technologies

Introduction to PySpark | Distributed Computing with Apache Spark

- Misc

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge

- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions

- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is Snowflake?

-
-
-

What is Snowflake?

Snowflake is a cloud-based data warehousing platform known for its scalability and flexibility. It separates storage and compute resources, enabling independent scaling and cost optimization. With a focus on ease of use, it facilitates secure data sharing and collaboration between organizations. Snowflake's cloud-native architecture leverages the benefits of providers like AWS, Azure, and Google Cloud. The platform offers built-in security features, including encryption and access control, making it suitable for organizations with stringent security and compliance needs. Overall, Snowflake provides a robust solution for storing, managing, and analyzing large datasets in the cloud.

What is Cloud Data Warehouse?

A cloud data warehouse is a type of data warehouse that is built, hosted, and managed on a cloud computing platform. Traditional data warehouses are on-premises solutions that require significant upfront investment in hardware, software, and maintenance. Cloud data warehouses, on the other hand, leverage the infrastructure and services provided by cloud service providers to offer a more scalable, flexible, and cost-effective solution.

What is Cloud Data Warehouse?

Popular examples of cloud data warehouses include Amazon Redshift, Google BigQuery, and Snowflake. Organizations often choose a cloud data warehouse based on their specific needs, existing cloud provider relationships, and the features offered by the platform.

Advantages of Snowflake

1. Cloud-Native Architecture: Snowflake is built for the cloud, making it easy to scale resources up or down based on demand. It leverages the elasticity and flexibility of cloud computing platforms like AWS, Azure, and Google Cloud.

Advantages of Snowflake

2. Separation of Storage and Compute: Snowflake separates storage and compute resources, allowing users to scale them independently. This architecture can lead to cost savings, as users only pay for the storage and compute resources they actually use.

Advantages of Snowflake

3. Data Sharing: Snowflake allows organizations to share data securely and easily with external parties without the need for complex data movement. This feature is particularly beneficial for collaboration and data exchange between organizations.

Advantages of Snowflake

4. Data Security: Snowflake provides robust security features, including data encryption, role-based access control, and audit logging. It complies with various industry standards and regulations, making it suitable for organizations with strict security requirements.

Advantages of Snowflake

5. Zero-Copy Cloning: Snowflake allows users to create clones of entire databases or specific tables without duplicating the data. This feature is useful for testing and development scenarios.

Advantages of Snowflake

6. Global Availability: With support for multiple cloud providers and regions, Snowflake offers global availability, enabling organizations to deploy their data warehouses close to their users for better performance.

Disadvantages of Snowflake

1. Cost: While Snowflake's separation of storage and compute can lead to cost savings, the overall cost can still be significant, especially for large-scale deployments.

Disadvantages of Snowflake

2. Learning Curve: Users will experience a learning curve when transitioning to Snowflake, especially if they are accustomed to traditional data warehousing systems.

Disadvantages of Snowflake

3. Dependency on Internet Connection: As a cloud-based solution, Snowflake relies on a stable internet connection. Organizations in regions with unreliable connectivity may face challenges.

Disadvantages of Snowflake

4. Limited Control Over Infrastructure: Some users may find the lack of control over the underlying infrastructure a drawback, especially for organizations with specific performance optimization requirements.

Disadvantages of Snowflake

5. Limited Advanced Analytics Capabilities: While Snowflake is excellent for traditional data warehousing and SQL-based analytics, it may not be as well-suited for advanced analytics or machine learning tasks compared to specialized platforms.

Disadvantages of Snowflake

6. Data Movement Costs: While data sharing is a strength, organizations should be mindful of potential costs associated with data movement, especially when transferring large volumes of data between different regions or cloud providers.

1. What is Snowflake in the context of Data Technology?

Snowflake is a cloud based data warehousing platform that allows organizations to store and analyze large volumes of data in a scalable and cost effective way.

2. How does snowflake differ from traditional data warehouse?

Unlike traditional data warehouses, Snowflake operates entirely in the cloud which offers on demand scalability, separation of storage and compute resources for more efficient data processing.

3. What is the significance of snowflake's Architecture?

Snowflake's Architecture is multi cluster, enabling parallel processing for faster query performance and allows users to scale resources independently to optimize costs.

4. How does Snowflake handle data sharing between organizations?

Snowflake's unique data sharing feature allows organizations to securely share live data with other Snowflake users, facilitating collaboration without the need for data movement or duplication.

5. What advantages does Snowflake offer in terms of data security?

Snowflake prioritizes data security through features such as end-to-end encryption, role-based access controls, and data masking, ensuring that sensitive information is protected at every level of the data warehouse.

5. What advantages does Snowflake offer in terms of data security?

Get 90% Course fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today. The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

5. What advantages does Snowflake offer in terms of data security?

H

5. What advantages does Snowflake offer in terms of data security?

- Geeks Premier League

- Geeks Premier League 2023

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python

- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP

- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems

- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner

- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming

- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

MS Excel Tutorial

K

MS Excel Tutorial

Microsoft Excel, often referred to as MS Excel, is a widely recognized spreadsheet program developed by Microsoft. It is available across multiple platforms, including Windows, macOS, Android, iOS, and iPadOS. It is an essential tool used for tasks such as data organization, formatting, calculations, data analysis, and more.

What You Will Learn from this Excel Tutorial

In this MS Excel tutorial, we cover everything from the basics to advanced-level topics, helping you progress from a beginner to an Excel master. The Excel tutorial is structured into three key sections: Beginner, Intermediate, and Advanced. In each section, you'll explore a variety of topics designed to boost your Excel skills. You will learn how to use functions, manage and sort data, conduct data analysis, create advanced charts and graphs, and much more.

What is Microsoft Excel?

As we know from the introduction part, MS Excel is a powerful spreadsheet program that is part of the Microsoft Office product group for business applications. It is an essential tool for various tasks such as data organization, calculations, data analysis, and visualization. Excel consists of a grid of cells arranged in rows and columns, where data is entered and stored. Each individual box within this grid is known as a cell.

What is Microsoft Excel?

Now, with its robust set of functions, users can perform complex calculations, analyze data using tools like pivot tables and sorting, and create charts or graphs to present data visually.

Advantages of Microsoft Excel

MS Excel is widely used for various purposes because the data is easy to save, and information can be added and removed without any discomfort or hard work.

Advantages of Microsoft Excel

- Easy To Store or Organize Data: Using Microsoft Excel one can easily store or organize large amounts of data.
- Easy To Recover Data: Because Excel is Microsoft product, hence the recovery of data is very easy.
- Easy to Manipulate or Visualization: Excel allows users to easily create data visualizations with charts.
- Direct Application of Mathematical Formulas: Using Excel one can perform mathematical

calculations on data sets, such as finding averages, standard deviations, and correlations.

- Free Template: Excel has many free templates to use.
- More Secure: Rather than other Spreadsheet programs Excel is more secure and less vulnerable.
- Clearer Visibility of Complex Information: Excel store all its data in tabular form hence, handling of complex data is easy

Microsoft Excel Quick Start Guide

The easiest way to get started with Excel is to use Microsoft 365, previously known as Office 365. It does not require downloading or installation of the program.

Excel Window Overview

Before you start using it, it's really important to understand what is where in the window. Some parts of the Excel window (like the Ribbon and Scroll bars) are standard in most Microsoft programs. However, there are other features that are more specific to spreadsheets, such as the formula bar, name box, and worksheet tabs.

Excel Window Overview

Excel Interface

What are Workbooks?

A collection of worksheets is referred to as a workbook (spreadsheet). Workbooks are your Excel files. There are various ways to begin working with an Excel workbook. Excel files are called workbooks. There are various ways to begin working with an Excel workbook. You can either start from scratch or use a pre-designed template to create a new workbook.

Creating and Opening Workbooks

Steps to Create a new blank workbook:

Creating and Opening Workbooks

Step 1: Select the File tab

Creating and Opening Workbooks

Step 2: ClickNew

Creating and Opening Workbooks

Step 3: ClickBlank Workbook

Creating and Opening Workbooks

Step 4:A new blank workbook will appear.

Creating and Opening Workbooks

You can also useTemplates.A template is a predesigned spreadsheet you can use to create a new workbook quickly. Templates often include custom formatting and predefined formulas.

Saving Workbook

Whenever we create a new workbook in Excel and insert the data into it, then we must save our workbook so that our data is not lost. As in previous versions of Excel, you can save fileslocallyto your computer You can also save your workbook to the cloud and also export your workbook with others.

Saving Workbook

- Save: When you save a file, you?ll only need to choose a file name and location the first time.

Then you can just use the save command to save it with the same name and location.

- Save As:When you use Save As, you?ll need to choose a different name and /or location for the copied version.

Saving Workbook

Note:Excel automatically saves your workbooks to a temporary folder while you are working on them. If you forget to save your changes or if Excel crashes, you can restore the file usingAutoRecover.

Saving Workbook

- Retrieving Data from Folders and Workbooks with Different Sheet Names
- Protecting Excel Worksheets and Workbooks
- How to Recover Unsaved Excel Files

Working with Worksheets in Excel?

A Worksheet is a collection of cells (basic data unit in the worksheet). Where you can store and manipulate data. Adding information to multiple worksheets simultaneously is also easily accomplished by grouping worksheets. In Excel, Worksheets can be easily added, renamed, and deleted.

Working with Worksheets in Excel?

When you open an Excel Workbook, there are three sheets by default and the name on them is sheet1, sheet2, and sheet3 you can change the name. You can perform multiple functions in Worksheets such as View a Worksheet, Select a Worksheet, Insert a Worksheet, Rename a Worksheet, and Delete a Worksheet.

Working with Worksheets in Excel?

- Introduction to Excel Spreadsheet
- How to Insert, Move, or Delete Page Breaks in Excel Worksheet?

Modifying Columns, Rows, and Cells in Workbook

Every row and column of a new workbook is set with the same height and width. Excel allows you to modify column width and row height in different ways including wrapping text and merging cells.

Modifying Columns, Rows, and Cells in Workbook

- How to Split Cells in Excel?
- CONCATENATE in Excel
- How to Insert a Picture in a Cell in MS Excel?

- ROWS and COLUMNS Functions in Excel With Examples
- How to Lock Cells For Editing in Excel?
- How To Concatenate Columns With Alt + Enter In Excel?
- ROWS and COLUMNS Functions in Excel With Examples

Inserting, Deleting, Moving, and Hiding in Excel Workbook

After working with a workbook for a while, you may want to insert new columns or rows, delete certain rows and columns, move them to a different location in the worksheet, or even hide them.

Inserting, Deleting, Moving, and Hiding in Excel Workbook

- How to Delete Blank Columns in Excel?
- Import Export of Data in Excel
- How to Delete All Rows Below Certain Row or Active Cell in Excel?
- How to Hide and Unhide Columns in Excel?
- How to Compare Two Columns and Delete Duplicates in Excel?
- How to Copy Cells on a Diagonal?
- Hide Error Values and Indicators in Cells in Excel

Inserting, Deleting, Moving, and Hiding in Excel Workbook

You can easily search your workbook using the Find feature, which also allows you to modify using the Replace feature.

Inserting, Deleting, Moving, and Hiding in Excel Workbook

Excel includes a Spell-check tool that ensures everything is spelled correctly in your workbook.

Inserting, Deleting, Moving, and Hiding in Excel Workbook

- How to Separate Numbers and Percentages from One Cell in Excel?
- How to Insert, Move, or Delete Page Breaks in Excel Worksheet?

Master Data Formatting in Excel?

In this section, we will read about data formatting in Excel. Data Formatting can be used to change the font size, To change the font, To change the font color, and many more.

Master Data Formatting in Excel?

- What is Data Formatting
- How to Insert Bullet Points in Excel?
- How to Expand Cells to Fit the Text Automatically in Excel?
- Dynamic Named Range in Excel
- File Formats in MS Excel
- How to Make and Format Reports in Excel?
- Richer Data Labels in MS Excel

Master MS Excel Formulas

The most important and useful feature of Microsoft Excel is to calculate numerical information using formulas. Excel can perform different calculations using mathematical operators (Addition, Subtraction, Multiplication, Division). We can insert a formula, create a formula, and edit a formula. Also, values can be modified on the basis of cell references.

Master MS Excel Formulas

- Excel Formulas
- Excel Formulas and Functions
- How to calculate the Sum and Average of numbers using formulas in MS Excel?
- How to Generate All Prime Numbers Between Two Given Numbers in Excel?
- How to Calculate Running Total in Excel?
- How to Correctly Sum Numbers With Units in a Range in Excel?
- Array Formulas in Excel
- How to Create an Array Formula in Excel?
- Organization Chart in Excel

- Dynamic Array Formulas in Excel
- Difference Between Array Formula and Regular Excel Formula
- Cell References in Excel ? Relative, Absolute, Mixed
- How to Format Time in Excel?
- Excel Formula to Get Next Day of Week
- How to Use Correlation in Excel?
- Sample Size Calculator With Excel
- How to Count the Number of Words in a Cell or a Range Cells in Excel?
- How to Calculate Conditional Probability in Excel?
- How to Calculate Partial Correlation in Excel?
- How to Calculate a Percentage of a Number in Excel?
- How to Calculate the Sum of Digits in a Number Using Non-Array Formulas in Excel?
- How to Apply a Percent Increase or Decrease to Values in Excel?
- How to Calculate Point-Biserial Correlation in Excel?
- How to Fix Excel Formulas that are Not Working properly?
- How to Use ChatGPT to Write Excel Formulas?
- How to Create Automatic Rolling Months in Excel?
- How to Create a Formula in Excel using Java?
- How to Calculate Fleiss' Kappa in Excel?

MS Excel Functions?

A function is a predefined formula that is used to perform calculations using specific values in a sequence. So, explore

MS Excel Functions?

- How to Insert a Function in Excel?
- New Functions in Excel 2019
- IF Function in Excel With Examples

- MID Function in Excel
- How to Apply a Percent Increase or Decrease to Values in Excel?
- Excel String Functions ? LEFT, RIGHT, MID, LEN, and FIND
- Statistical Functions in Excel With Examples
- Excel Date Functions ? with Formula Examples
- INDEX and MATCH With Multiple Criteria In Excel
- How to Create a Custom Date Format in Excel?
- INDEX and MATCH Function in Excel
- CHOOSE Function in Excel
- How to VLOOKUP to Return Multiple Values in One Cell in Excel?
- Excel VLOOKUP with Dynamic Column Reference
- Why INDEX MATCH is Better Than VLOOKUP in Excel?
- XLOOKUP Function in Excel With Examples
- XLOOKUP vs INDEX-MATCH in Excel
- How to Use the Excel STDEV.P Function?

Basic Tips For Working with Data in Excel

Excel Workbooks are designed to store a lot of information no matter you are working with 10 cells or 10,000 cells. Excel has many features to help you Organize your data and Find what you need.

Basic Tips For Working with Data in Excel

Sometimes you may want to see certain rows and columns all the time in your worksheet, mainly Header cells. By freezing rows and Columns in place, you'll be able to scroll through your content.

Basic Tips For Working with Data in Excel

- Excel Shortcuts
- Top 20 Excel Shortcuts That You Need To Know
- 6 Tips for Creating Effective Data Visualizations

- How to Compare Two Columns in Excel?
- How to Create a Dependent Drop Down List in Excel?
- How to Count Duplicate Values in a Column in Excel?
- How to Compare Two Lists or Datasets in Excel?
- How to Automate Data Entry In Excel?
- How to average a range of data ignoring zero in Excel?
- How to Separate Text and Numbers from One Cell Into Two Columns?
- How to Remove Text Before or After a Specific Character in Excel?
- How to Add or Subtract Dates in Excel?
- How to Convert Weekday Name to Number in Excel?
- How to Hide the Zero Values in Excel?
- How to average a range of data ignoring zero in Excel?
- How to Normalize Data in Excel?
- How to Extract the last N words in Excel?
- How to Apply the Empirical Rule in Excel?
- How to Create a Custom Date Format in Excel?
- How to Use SQL Statements in MS Excel?

1. Sorting Data

You can quickly reorganize your data in a sequence you want in your worksheet by sorting your data. You can sort your data alphabetically, numerically, and in many other ways.

1. Sorting Data

- How to Sort by the Last Name in Excel?
- How to Sort Alphabetically in Excel?
- How to Sort by Number in Excel?
- How to Sort by Date in Excel?
- How to Sort Data by Color in Excel?

- How to Perform Multi-Level Data Sorting in Excel?
- Sorting and Adding Subtotals to Excel

2. Filtering Data

Filters can be applied to data in your worksheet allowing you to view only the information you need. Formatting Data as a Table can improve the overall look of your workbook and help you to organize your content and make your data easier to use.

2. Filtering Data

- How to Filter Data in Excel?
- How to add Filters in MS Excel?
- How to Count Duplicate Values in a Column in Excel?
- How to Delete Rows Not Shown in Filter?
- Dynamic Excel Filter Search Box

3. Groups and Subtotals

Worksheets with a lot of content can sometimes feel overwhelming. Excel provides you with the feature of organizing data into Groups, allowing you to easily show and hide different sections of your worksheet. The Subtotal command allows you to quickly summarize data. Below are some functions that you can learn in Groups and subtotal.

3. Groups and Subtotals

- To group rows or columns
- To hide and show groups

3. Groups and Subtotals

- How To Concatenate Columns With Alt + Enter In Excel?
- How to Find Duplicates in Excel
- How to Group Adjacent Columns or Rows Separately or Independently in Excel?

- Compare Data in an Excel Chart using Drop Down Lists

Working with MS Excel Tables

After entering your data in the worksheet, you may want to format your data as a table. They will help to organize your data content and make your data easier to use and understand. Excel has many predefined table styles, allowing you to create tables quickly and easily.

Working with MS Excel Tables

- How to Create Relational Tables in Excel?
- How to Create a Data Table In Excel?
- How to Create a Contingency Table in Excel?
- How to Create a Two-Variable Data Table in Excel?
- How to Make a Calendar in Excel?

Modifying Tables

It is easy to modify the look of the table after adding it to a worksheet. Excel includes several options for customizing a table, including adding rows or columns, changing the table style, modifying the table style options, and removing a table.

Formatting Data as a Table

Step 1: Select the cells you want to format as a table

Formatting Data as a Table

Step 2: Click the Format as Table in the styles group from the Home tab

Formatting Data as a Table

Step 3: Now select Table Style from the drop-down menu

Formatting Data as a Table

Step 4: Confirm the cell range for the table

Formatting Data as a Table

Step 5: Click OK if your table has headers

Formatting Data as a Table

Step 6: The cell range will be formatted in the selected table style.

MS Excel Charts

It is difficult to examine Excel workbooks that contain a lot of data. Charts help you to illustrate your workbook data graphically, which makes it easy to examine and compare the Excel workbook and make better decisions.

MS Excel Charts

You can learn how to insert and modify Excel Charts to visualize comparisons in your data. There are many options to change the chart and layout style.

MS Excel Charts

- How to Create Charts in Excel?
- How to Move and Resize a Chart in Excel?
- Types of Charts in Excel
- How to Change Chart Style in Excel?
- Formatting Charts in Excel
- How to Add a Horizontal Line in a Chart in Excel?
- How to Add Lines In An Excel Clustered Stacked Column Chart?
- How to Create an X-Y Scatter Plot in Excel?
- How to Make a Bar Graph in Excel?
- How to Create a Column Chart in Excel?
- How to Create Line Charts in Excel?
- How to Create a Waterfall Chart in Excel?
- How to Make a Comparison Chart in Excel?

- How to Create an Excel Step Chart Formula Using the Small Function?
- How to Create a Chart from Multiple Sheets in Excel?
- How To Create A Pictograph In Excel?
- Using Columns and Bars to Compare Items in Excel Chart
- Organization Chart in Excel
- How to Calculate Cumulative Frequency table in Excel?
- How to Graph three variables in Excel?
- How to Update, Change and Manage the Data in a Chart in Excel?
- How to Auto Update a Chart After Entering New Data in Excel?
- Actual Vs Target Chart In Excel With Floating Markers
- Dynamic Organizational Chart in Excel
- How to Create Dynamic Chart Titles In Excel?
- How to Create a Goal Line on a Chart in Excel?
- How to Create Advanced Charts and Graphs in Excel?
- How to Graph Three Sets of Data Criteria in an Excel Clustered Column Chart?
- How to Make a Frequency Polygon in Excel?
- How to Find the Intersection of Two Curves in Excel?
- How to Create a Sales Funnel Chart In Excel?
- How to make Overlapping Bar Chart in Excel?
- How to Make a Dynamic Gantt Chart in Excel?
- How to Create a Rolling Chart in Excel?
- Scatter and Bubble Chart Visualization in Excel
- Modifying Data Range in Excel Charts
- How to Plot Bivariate Data in Excel?
- Compare Data in an Excel Chart using Drop Down Lists
- How to Create a Survival Curve in Excel?
- Multiple Series in One Excel Chart

- How To Make A Graph With Multiple Axes With Excel?

Understanding Conditional Formatting

Similar to charts, Conditional formatting provides a way to visualize data and make worksheets easier to understand. It allows you to automatically apply formattings- such as formattings, and data bars to one or more cells. You can use Conditional formatting by using the below steps.

How to use Conditional Formatting in Excel

Step 1: Select desired cells

How to use Conditional Formatting in Excel

Step 2: Click on the Conditional formatting command from the Home Tab

How to use Conditional Formatting in Excel

Step 3: Select the desired rule from the menu that appears

How to use Conditional Formatting in Excel

Step 4: Enter the desired values in the dialog box

How to use Conditional Formatting in Excel

Step 5: Select Formatting styles.

How to use Conditional Formatting in Excel

You can also use Conditional Formatting presets and Remove the Conditional Formatting.

How to use Conditional Formatting in Excel

- How to Count Unique Values in Excel?
- How to Remove Hyperlinks in Excel?
- How to Remove Blank Rows in Excel?
- How to Turn On/Off Scroll Lock in Excel?
- How to Apply Conditional Formatting Based On VLookup in Excel?

What are Pivot Tables in Excel?

If your workbook contains lots of data, it can sometimes be difficult to analyze all the information in your worksheet. Pivot Tables can help to make worksheets more manageable by summarizing data and allowing you to manipulate it in different ways.

How to Create Pivot Tables

Step 1: Select the data cells

How to Create Pivot Tables

Step 2: From the Insert tab click on the PivotTable

How to Create Pivot Tables

Step 3: Choose your settings in the Create PivotTable Dialog box and click OK

How to Create Pivot Tables

Step 4: A blank PivotTable and Field List will appear in a new worksheet.

How to Create Pivot Tables

- Prepare Source Data for Pivot Tables In MS Excel
- How to Create a Power Pivot Table in Excel?
- How to Prevent Grouped Dates In Excel Pivot Table?
- How to Hide Zero Values in Pivot Table in Excel?
- How to Remove Old Row and Column Items from the Pivot Table in Excel?
- Exploring Data with Pivot Tables in Excel
- Sorting Pivot Table in Excel
- How to Flatten Data in Excel Pivot Table?
- How to Remove Pivot Table But Keep Data in Excel?
- How to Add and Use an Excel Pivot Table Calculated Field?

What is Excel VBA and Macros in Excel

In this section we will deep dive into the MS Excel VBA and Macros both of the feature allow users to automate repetitive task. So, explore the section to know what is Macros and VBA and how it works.

What is Excel VBA and Macros in Excel

- How to Configure Macros in Excel?
- How to Run a Macro
- Relative References in Excel Macros
- Absolute References in Excel Macros
- How to Password Protect Your Excel Macro?
- Assigning Excel Macro to Objects
- Editing Excel Macros in Excel
- How to Run a Macro Automatically When Workbook Opens in Excel?
- UserForms in Excel Macros
- Excel Personal Macro Workbook
- How to Enable a Macro
- How to Record a Macro

VBA in Excel

- How to Insert and Run VBA Code in Excel?
- Get, Set, or Change Cell value in Excel VBA
- How to Run Code from a Module in Excel VBA?
- How to Delete a Module in Excel VBA?
- Workbook and Worksheet Object in Excel VBA
- Excel VBA Events
- Basic Object Model in Excel VBA
- Variables and Data Types in VBA Excel
- Application Objects in Excel VBA

- VBA Objects in Excel
- VBA Strings in Excel
- VBA Constants in Excel
- VBA Arithmetic Operators in Excel
- VBA Arrays in Excel
- VBA Print Statement in Excel
- VBA Find Function in Excel
- VBA Date and Time Functions in Excel
- VBA Collections in Excel
- How to Set Variable to Cell Value in Excel VBA?
- VBA Subroutine in Excel ? How to Call Sub in VBA?
- Sub Procedure in Excel VBA
- Excel VBA Error Handling
- Generating Dynamic Charts With VBA in Excel
- Debugging VBA Code in Excel
- How to Declare and Initialize String Array in Excel VBA?
- How to Add a Comment in a VBA in Excel?
- Multidimensional Arrays in Excel VBA
- How to Convert VBA Collections to Array in Excel?
- How to Round With Doubles in Excel VBA?
- Trapping Dynamic Ranges in Excel VBA
- InputBox Function in Excel VBA
- MessageBox in Excel VBA
- How to Get a List of User Defined Functions in Excel VBA?
- ActiveX Control in Excel VBA
- How to Create an Input Box With Multiple Inputs in Excel Using VBA?
- How to Create Charts in Excel Using Worksheet Data and VBA?

- Programming Charts in Excel VBA

Comments and Co-authoring

These are the two most powerful features that allow you to work with others on the same spreadsheet: Comments and Co-authoring. These features come in the role when you're working on a workbook and find that you need the help of others.

Comments and Co-authoring

The Track Changes feature can also be helpful to review changes before making them permanent. It's still available in Office 365, but it's now hidden by default.

Inspecting and Protecting Workbooks

While sharing a workbook in Excel, everyone wants to make sure it doesn't include any spelling errors or information you want to keep private. Excel includes several tools to help finalize and protect your workbook, including Document Inspector and Protect Workbook Feature.

Inspecting and Protecting Workbooks

- Protect a Sheet in MS Excel
- How to Manage Passwords Using MS Excel?
- Protecting Excel Worksheets and Workbooks
- Protecting Excel Workbook using Automation Anywhere

What are Reference Styles in Microsoft Excel

Every Excel spreadsheet contains rows and columns. Most of the time columns are denoted by letters (A, B, C) and rows are denoted by numbers (1, 2, 3). This is known as the A1 reference style.

What are Reference Styles in Microsoft Excel

Whether you use a digital pen, touchscreen, or mouse, the drawing feature in the office can help you to add notes, create shapes, edit text, and more.

What are Reference Styles in Microsoft Excel

If you need graphics for the project, there is a feature you can use called icons. Icons are a library of modern, professional graphics included with Office 365 and 2019, and they can be customized to fit your needs.

What are Reference Styles in Microsoft Excel

- Excel's CHOOSE Function with Array
- Handling Integers in Advanced Excel
- Managing External Data Connection in Advanced Excel
- Power Query ? Source Reference as File Path in Cell
- How to Merge Content of All Files in Folder with Power Query?
- How to Copy Power Query Queries in Excel?

Data Analysis, Automation in MS Excel

This section highlights advanced functionalities, transforming MS Excel into a powerful tool for data analysis, automation, and efficient workflow management.

Power View

Power View is the data visualization technology that lets you create interactive charts, graphs, maps, and other various visuals. Power View enables interactive data exploration that encourages intuitive ad-hoc reporting. Power view is completely based on the Data model in your workbook. There are various functions and operations to be performed on Power View listed below.

Power View

- How to Enable and Use Power View in Excel?
- How to Insert or Show Power View in Excel?
- Data Visualizations in Power View
- Chart Visualizations in Excel Power View
- Table Visualization in Excel Power View

- Card Visualization in Excel Power View
- Bar Chart Visualization with Excel Power View
- Matrix Visualization in Excel Power View
- Multiple Visualizations in Excel Power View
- Line Chart Visualization in Excel Power View
- Tiles Visualization with Excel Power View
- Exploring Data with Power View Maps in Excel
- Exploring Data with Power View in Excel

Power Pivot

It is an Excel add-in that can be used to perform better data analysis and create sophisticated data models. With Power Pivot, you can work on large volumes of data from different sources, and perform information analysis effectively. When you have multiple tables, Power Pivot can help you to link them together. After linking them together you can create a Pivot table that will help you to view the data in a single table.

Power Pivot

- Power Pivot for Excel Tutorial
- Excel Power Pivot ? Managing Data Model
- Creating a Data Model in Excel
- How to Flatten Data in Excel Pivot Table?
- Features of Excel Power Pivot
- Hierarchies in Excel Power Pivot
- Exploring Data with Excel Power Pivot
- Table and Chart Combinations in Excel Power Pivot
- Creating a Power Pivot Chart in Excel
- Loading Data with Power Pivot in Excel

Advanced Excel

Advanced Excel provides a good insight into the latest and advanced features available in Microsoft Excel.

Advanced Excel

- Instant Data Analysis in Advanced Excel
- Dynamic Map with Drop-Down in Excel
- Handling Integers in Advanced Excel
- Managing External Data Connection in Advanced Excel
- Advanced Excel ? Chart Design

Power Query

It is a tool available in Excel that helps you to import data from many different sources and then clean, transform and reshape the data as required. You need to set up a query once and then reuse it with a simple refresh. The query can import millions of rows into the data model for analysis.

Power Query

- Power Query ? Source Reference as File Path in Cell
- Connecting Excel to SQLite
- How to Copy Power Query Queries in Excel?
- How to Import, Edit, Load and Consolidate Data in Excel Power Query?
- How to Create a Relative File Path with Power Query in Excel?
- Text Modifications with Power Query in Excel
- Date Modifications with Power Query in Excel
- How To Get Current Date in Power Query?
- Excel ? Types of Merging of Queries in Power Query
- Power BI ? Tools and Functionalities
- Power BI ? Data Sources and its type

What-if Analysis

This feature can help you experiment and answer questions with your data, even when the data is incomplete. To learn how to use the What-if analysis, use a tool called Goal Seek.

What-if Analysis

- Instant Data Analysis in Advanced Excel

Additional MS Excel Tips

- How to Use the Data Consolidation Feature in Excel?
- How to use Autofill Feature in MS Excel?
- Inquire Feature in Excel

Why to Learn Excel

Here in this section, you will get to know why this MS Excel tutorial is needed and what are the benefits.

Why to Learn Excel

- Enhanced Data Analysis: Excel empowers users to analyze large datasets, identify trends, and draw meaningful insights with its powerful functions and tools.
- Increased Efficiency: Excel's automation features streamline repetitive tasks, saving time and reducing errors.
- Improved Decision Making: Excel's clear and visually appealing data organization enables users to make informed decisions in various contexts.
- Advanced Financial Analysis: Excel's financial functions and modelling capabilities are essential for complex calculations and evaluating investment opportunities.
- Versatility and Integration: Excel seamlessly integrates with other Microsoft Office applications and external data sources, facilitating data sharing and analysis.
- Career Advancement: Proficiency in Excel opens doors to job opportunities requiring data analysis, reporting, and financial management skills, enhancing professional value.

Microsoft Excel Features

Anyone can learn how to use and take advantage of the program's powerful features.

Microsoft Excel Features

- AutoFormat: Use predefined table formatting settings to easily format data.
- AutoSum: Automatically calculates the sum of a row or column with a simple addition formula.
- List AutoFill: Automatically formats cells when a new component is added to the end of a list.
- AutoFill: Quickly fill cells with repetitive or sequential records like dates, numbers, or text.
- AutoShapes: Create geometric shapes, arrows, flowcharts, and other objects for graphs and diagrams.
- Wizard: Provides helpful recommendations and drag-and-drop features for more efficient workflow.
- Charts: Display data in visual formats like Pie, Bar, and Line charts to enhance understanding.
- PivotTable: Easily summarize, analyze, and visualize complex data with just a few clicks.
- Shortcut Menus: Speed up tasks by using shortcut commands, saving time on lengthy processes.

Excel Can Be Used For

- Analysis
- Data Entry
- Data Validation
- Accounting
- Budgeting
- Data analysis
- Visuals and Graphs
- Programming
- Financial Modeling

Microsoft Excel Keyboard Shortcuts

Knowing keyboard shortcuts can significantly speed up your Excel workflow. Here are a few

commonly used shortcuts:

Microsoft Excel Keyboard Shortcuts

- Ctrl + C: Copy selected cells
- Ctrl + V: Paste copied cells
- Ctrl + Z: Undo the last action
- Ctrl + S: Save the workbook
- Ctrl + F: Open the Find dialog box
- Ctrl + B: Apply bold formatting
- Ctrl + Shift + L: Apply filters to a table

Microsoft Excel Keyboard Shortcuts

More MS Excel Keyboard Shortcuts ?Excel Keyboard Shortcuts

Extra Features in MS Excel

There are a couple of new chart types in Excel: Map charts and Funnel charts.

Extra Features in MS Excel

- Dynamic Map with Drop-Down in Excel
- How to Create Dynamic Excel Dashboards Using Picklists?

Excel Tutorial PDF Download

Download Complete Excel Tutorial in PDF ?Click Here

Conclusion

Mastering Excel is a powerful step toward improving productivity and data management. Whether you're analyzing data, creating visually compelling charts, or automating tasks with functions and formulas, Excel offers versatile tools to meet your needs. We are sure that by now, this MS Excel tutorial will help you gain a solid foundation in navigating Excel's interface, using formulas, managing data, and creating visualizations.

Q1: What are the benefits of using MS Excel?

The main benefit of using MS-Excel are:

Q1: What are the benefits of using MS Excel?

- The MS-Excel is best way to store a huge amount of data.
- You can perform different Calculations.
- There are several tools that are available for Data analysis
- You can print reports easily.
- You can automate your code easily .
- You can Transform your data.
- It's easy to reformat and rearrange your data.
- Process data and analysed with graphs and charts

Q2: What are the shortcut keys of MS Excel?

Shortcut Keys are used for navigating and working with Excel spreadsheets with speed. The main shortcut keys of MS-Excel are:-

Q2: What are the shortcut keys of MS Excel?

- Key to open a workbook (Ctrl+N)
- Key to open a saved workbook(Ctrl+O)
- Key to save a workbook(Ctrl+S)
- Key to copy the selected cells(Ctrl+C)
- Key to paste the copied cells(Ctrl+V)
- Key to cut the selected cells(Ctrl+X)
- Key to close the workbook(Ctrl+W)
- Key to remove all the contents from the cell(Delete)
- Key to print the workbook(Ctrl+P)
- Key to undo(Ctrl+Z)

Q3: Is MS Excel an Operating system?

MS-Excel is not an operating system. It is a spreadsheet program included in the Microsoft Office suite of several applications. It is introduced by Microsoft and it is developed for Windows, Android, macOS, and iOS users.

Q4: What do you mean by Cell address?

Cell address refers to the address that is obtained by the combination of the row number and column alphabet. Every cell has a distinct cell address.

Q5: How can you protect your workbooks in Excel?

To protect your workbook below are three ways:

Q5: How can you protect your workbooks in Excel?

- Set a password to open the workbook.
- Protect the sheet from being added, deleted, hidden, or unhidden.
- Protecting window sizes or positions from being changed.

Q5: How can you protect your workbooks in Excel?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Q5: How can you protect your workbooks in Excel?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Q5: How can you protect your workbooks in Excel?

- Excel
- Microsoft Office
- News
- Excel Roadmap

Q5: How can you protect your workbooks in Excel?

K

Q5: How can you protect your workbooks in Excel?

[Read More](#)

Q5: How can you protect your workbooks in Excel?

N

Editing Excel Macros in Excel

S

Best Microsoft Excel High-Paying Jobs with Excel Skills in 2024

S

AI in Excel: Top 20 Excel AI Tools in 2025 (Free/Paid)

K

Excel VLOOKUP Function - Excel Guide for Beginners

A

Scenario Manager in Excel

A

Excel VBA | count() functions

S

How to Sum Diagonal Cells in a range in Excel?

A

Advanced Excel - Chart Design

A

How to Use for Each Loop in Excel VBA?

A

How to Hide Zero Values in Excel

How to Hide Zero Values in Excel

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

How to Hide Zero Values in Excel

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos

- Geeks Community

How to Hide Zero Values in Excel

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

How to Hide Zero Values in Excel

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

How to Hide Zero Values in Excel

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning

- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

How to Hide Zero Values in Excel

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

How to Hide Zero Values in Excel

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

How to Hide Zero Values in Excel

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

How to Hide Zero Values in Excel

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

How to Hide Zero Values in Excel

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD

- System Design Bootcamp
- Interview Questions

How to Hide Zero Values in Excel

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

How to Hide Zero Values in Excel

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

How to Hide Zero Values in Excel

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL

- MongoDB

How to Hide Zero Values in Excel

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

How to Hide Zero Values in Excel

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

How to Hide Zero Values in Excel

- More Tutorials
- Software Development
- Software Testing
- Product Management

- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

How to Hide Zero Values in Excel

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

How to Hide Zero Values in Excel

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

How to Hide Zero Values in Excel

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course

- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

How to Hide Zero Values in Excel

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

How to Hide Zero Values in Excel

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT

- Data Science Course with IBM Certification

How to Hide Zero Values in Excel

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

How to Hide Zero Values in Excel

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

How to Hide Zero Values in Excel

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

How to Hide Zero Values in Excel

Introduction & Setup the Power BI

- What is Business Intelligence?
- Power BI - Rise of Microsoft Power BI as a Data Analytics powerhouse
- Power BI - Key Components
- Power BI - Practical Applications

- How to Install Power BI on Windows?
- Power BI - Data Sources and its type
- Power BI - Differences Between Microsoft Power BI and SSRS
- Power BI Free vs Power BI Pro vs Power BI Premium

The Power BI Query Editor

- Power BI - How to edit in Power BI App?
- Power BI - DAX Date Functions
- Power BI - How to Add Conditional Columns?
- DAX Aggregate Functions in Power BI
- Power BI - Excel Integration

PowerBI Dash Board Visualisation

- Power BI - Dashboard Introduction
- Power BI - Dashboard Actions
- Power BI - Create a Stacked Bar Chart
- Power BI - How to add Reports to Dashboards?
- Power BI - Report Level Filters
- Power BI - How to Format Matrix?
- Power BI - Slicer
- Power BI - How to Format a Card?
- Power BI - Format Multi-Row Card
- Power BI - How to Create a Map?
- Power BI - How to Format Map
- Power BI - How to Create a Treemap?
- Power BI - How to Format Tree Map?
- Connect Power BI Desktop with Power BI Service
- Power BI - Create a Table

- Power BI - How to Format Table?
- Power BI - Create a Stacked Column Chart
- Power BI - How to Format Stacked Column Chart?
- Power BI - Create 100% Stacked Column Chart
- Power BI - Create 100% Stacked Bar Chart
- Power BI - Format Stacked Bar Chart
- Power BI - How to Create a Stacked Area Chart
- Power BI - Format Area Chart
- Power BI - Create a Radial Gauge Chart
- Power BI - Explain Key Performance Indicators (KPIs) dashboards

Power BI _ Format KPIs Chart

- Power BI - Format Clustered Bar Chart

Power BI _ Format Clustered Bar Chart

- Power BI - How to Create a Waterfall Chart?
- Power BI - Format Waterfall Chart
- Power BI - Create a Filled Map
- Power BI - Format Filled Map
- Power BI - How to Create a Scatter Chart?
- Power BI - Format Scatter Chart

Showing Trends with Line Charts

- Power BI - Format Line Chart
- Power BI - How to Create a Shape Map?
- Power BI - Format Donut Chart
- Power BI - Format Pie Chart
- Power BI - Format Ribbon Chart

- Power BI - Create a R Script Visual
- Power BI - Format Line and Stacked Column Chart
- Power BI - Format Line and Clustered Column Chart

DAX Introduction

- Power BI - How to Create Calculated Columns?
- Power BI- Functions
- Power BI- DAX Window Function
- Power BI - DAX Logical Functions
- Power BI - DAX Information Functions
- Power BI - DAX TEXT Functions
- Power BI- DAX Index Function
- Power BI - DAX Trigonometric Functions
- Power BI - DAX COUPDAY Financial Function
- Power BI - DAX Depreciation Functions
- Power BI - Distinct() function
- Power BI - DAX Math Functions
- Power BI - DAX Counting Functions
- Power BI- DAX Bitwise Functions

Power BI Tutorial | Learn Power BI

-
-
-

Power BI Tutorial | Learn Power BI

Power BI is a Data Visualization and Business Intelligence tool by Microsoft that converts data from different data sources to create various business intelligence reports. Microsoft Power BI makes it

easy for businesses to spot trends, track performance, and make data-driven decisions.

Power BI Tutorial | Learn Power BI

This Power BI tutorial is your one-stop guide for learning Power BI from Scratch. It covers all the basic and advanced concepts of Power BI like, filter Power BI, Power BI desktop, functions in Power BI, Power BI services, etc.

Power BI Tutorial | Learn Power BI

Whether you are a beginner or an experienced professional, this free Power BI tutorial will help you learn about all the Important concepts of Power BI with ease.

Power BI Tutorial | Learn Power BI

PowerBI Tutorial

What is Business Intelligence

Business intelligence is the process of analyzing business data and presenting it in user-friendly graphical ways such as dashboards, reports, charts, graphs, etc. These reports help the business owners to drive changes, eliminate inefficiencies, and adapt to the market changes. There are many business intelligence (BI) tools that help in building business intelligence reports.

What is Business Intelligence

Nearly 50% of companies use BI tools, and many companies are opting to integrate BI tools into their system. Tools like Microsoft Power BI are used by over 3000 companies for business intelligence. Let's discuss more about Power BI.

What is Power BI

Power BI is the collection of software apps, services, and connectors that turn the raw data into meaningful, immersive, and interactive insights.

What is Power BI

It was developed by Microsoft and is very versatile to use. It can accept input from Excel sheets, databases, or even cloud storage. You can connect the data sources, derive insights, and share them with others. Power BI is the most popular business intelligence tool out there, but why? Let's discuss why is Power BI in demand and what makes it different from other tools.

Why Power BI?

In today's world, data is the new oil. Organizations need a tool that can help them understand the large amount of data that they are collecting. Microsoft Power BI was introduced to fulfill their needs. It is a powerful data visualization and analysis tool that allows businesses to turn raw data into actionable insights and reports.

Why Power BI?

- Power BI helps users who are beginners who do not come from an analytical or tech background. Still, they can create reports on the Power BI desktop.

Why Power BI?

Microsoft Power BI comes with a free or paid version. The free version only provides Power BI tools like Power BI Desktop and Power Q&A to dashboards. Whereas, in the Pro version they provide services like live report sharing, Power View, and more Power BI apps.

Why Power BI?

Now that we are familiar with Power BI, let's see how to use Power BI by installing it on our devices.

Microsoft Power BI Installation

Power BI Desktop is available in 32-bit and 64-bit versions. The Power BI desktop app is used to create reports, while Power BI Services (Software as a Service ? SaaS) is used to publish the reports, and the Power BI mobile app is used to view the dashboards and reports.

Microsoft Power BI Installation

Download Here:<https://www.microsoft.com/en-us/download/details.aspx?id=45331>

Microsoft Power BI Installation

After the successful installation of Power BI, you can start with this tutorial and learn how to use Power BI.

Power BI Tutorial For Beginners

Here is the Power BI tutorial specially designed for beginners. The tutorial is divided into 5 sections and each section provides you the necessary materials to progressively learn Power BI. As you complete each section you move forward to your goal of becoming a Power BI specialist.

Section 1: Power BI ? Introduction & Setup

- What is Business Intelligence?
- What is Power BI and Why Power BI in Data Analysis
- Power BI Components
- Practical Applications
- Downloading Power BI & Adjusting Settings
- Advantages of Power BI, Disadvantages of Power BI
- Types of Data Connectors in Power BI Desktop
- Differences Between Microsoft Power BI and SSRS
- Power BI Free vs Power BI Pro vs Power BI Premium

Section 2: Power BI ? Query Editor

- Edit PowerBI App
- Query Editor in Power BI for Data Transformation
- Working with Numbers in Power BI
- Working with Date & Time Tools
- Creating a Rolling Calendar in Power BI
- Conditional Columns in Power BI

- Grouping & Aggregating Records
- Merge and Append Queries in Power BI
- Manage data source settings and permissions
- Data refresh in Power BI
- Power BI Data Types
- Explain Power BI Hierarchy, and How to Use it
- Power BI ? Excel Integration

Section 3: Power BI DashBoard and Power BI Visualisation

- Power BI ? Dashboard
- Power BI ? Dashboard Actions
- Adding Objects to the Power BI Report Canvas and Exploring the ?Report? View
- Creating Table Relationships & Data Models in Power BI
- Inserting Basic Charts & Visuals in Power BI
- Conditional Formatting
- How to add Reports to Dashboards
- Power BI Report Formatting Options
- Power BI Report Filtering Options
- Exploring Data with Matrix Visuals
- Filtering with Date Slicers
- Adding Trend Lines & Forecasts
- Editing Power BI Report Interactions
- Adding Drillthrough Filters
- Inserting Text CardsHow to Format a Card?Format Multi-Row Card
- How to Format a Card?
- Format Multi-Row Card
- Visualizing Geospatial Data with MapsHow to Format Map

- How to Format Map
- Visualizing Data with TreemapsFormat Tree Map
- Format Tree Map
- Managing & Viewing Roles in Power BI Desktop
- Creating a Simple TablePower BI ? Format Table Chart
- Power BI ? Format Table Chart
- Create a Stacked Column ChartPower BI ? Format Stacked Column Chart
- Power BI ? Format Stacked Column Chart
- Create a 100% Stacked Column ChartFormat 100% Stacked Bar Chart
- Format 100% Stacked Bar Chart
- Create a Stacked Bar ChartPower BI ? Format Stacked Bar Chart
- Power BI ? Format Stacked Bar Chart
- How to Create a Stacked Area ChartPower BI ? Format Area Chart
- Power BI ? Format Area Chart
- Create a Radial Gauge Chart
- Create Key Performance Indicators (KPIs) ChartPower BI ? Format KPIs Chart
- Power BI ? Format KPIs Chart
- Format Clustered Bar ChartPower BI ? Format Clustered Bar Chart
- Power BI ? Format Clustered Bar Chart
- How to Create a Waterfall Chart?Format Waterfall Chart
- Format Waterfall Chart
- Create a Filled MapFormat Filled Map
- Format Filled Map
- Create a Scatter Chart.Format Scatter Chart
- Format Scatter Chart
- Showing Trends with Line ChartsPower BI ? Format Line Chart
- Power BI ? Format Line Chart

- How to Create a Shape Map?
- Format Donut Chart
- Format Pie Chart
- Format Ribbon Chart
- Create an R Script Visual
- Line and Stacked Column chart
- Format Line and Clustered Column Chart

Section 3: Power BI Dashboard and Power BI Visualisation

- How to Format a Card?
- Format Multi-Row Card

Section 3: Power BI Dashboard and Power BI Visualisation

- How to Format Map

Section 3: Power BI Dashboard and Power BI Visualisation

- Format Tree Map

Section 3: Power BI Dashboard and Power BI Visualisation

- Power BI ? Format Table Chart

Section 3: Power BI Dashboard and Power BI Visualisation

- Power BI ? Format Stacked Column Chart

Section 3: Power BI Dashboard and Power BI Visualisation

- Format 100% Stacked Bar Chart

Section 3: Power BI Dashboard and Power BI Visualisation

- Power BI ? Format Stacked Bar Chart

Section 3: Power BI Dashboard and Power BI Visualisation

- Power BI ? Format Area Chart

Section 3: Power BI DashBoard and Power BI Visualisation

- Power BI ? Format KPIs Chart

Section 3: Power BI DashBoard and Power BI Visualisation

- Power BI ? Format Clustered Bar Chart

Section 3: Power BI DashBoard and Power BI Visualisation

- Format Waterfall Chart

Section 3: Power BI DashBoard and Power BI Visualisation

- Format Filled Map

Section 3: Power BI DashBoard and Power BI Visualisation

- Format Scatter Chart

Section 3: Power BI DashBoard and Power BI Visualisation

- Power BI ? Format Line Chart

Section 4: DAX Introduction

- Data Analysis Expressions (DAX)
- Intro to DAX Calculated Columns
- Creating measures using DAX
- Adding Columns & DAX Measures in Power BI Desktop
- Filter Context in Power BI
- Common DAX Function Categories
- Basic Date & Time Functions
- DAX Window Function
- Conditional&Logical Functions

- DAX Information Functions
- DAX Text Functions
- Index function in Power BI
- DAX Trigonometric Functions
- DAX COUPDAY Financial Function
- DAX Depreciation Functions
- Power BI ? Distinct() function
- Basic Math in Power BI
- Power BI ? DAX Trigonometric Functions
- COUNT Functions
- Power BI ? DAX Date Functions
- DAX Aggregate Functions in Power BI
- Power BI ? DAX TEXT Functions
- Power BI ? DAX Depreciation Functions
- Power BI- DAX Bitwise Functions

Section 5: Creating Table Relationships & Data Models in Power BI

- What is a ?Data Model??
- Principles of Database Normalization
- Understanding Data Tables vs. Lookup Tables
- Understanding Table Relationships vs. Merged Tables
- Creating Table Relationships in Power BI Desktop
- Managing & Editing Table Relationships | Power BI
- Managing Active vs. Inactive Relationships | Power BI
- Connecting Multiple Data Tables in Power BI
- Understanding Filters in Power BI
- Hiding tables, columns, and fields from Power Pivot

Features of Power BI

Some most important Power BI features are given below :

Features of Power BI

- Data Visualization: Power BI is used by organizations to create interactive reports and dashboards that visualize data in various ways, including charts, graphs, and maps.
- Collaboration: Power BI allows multiple users to work on the same report or dashboard at the same time. This makes it easy for teams to collaborate, share data and share insights.
- Data Transformation: Power BI can transform raw data into useful information that is a more usable format by cleaning and manipulating data. This allows businesses to work with clean, accurate data that is easy to analyze.
- Integration: Power BI integrates with a wide range of other Microsoft products or applications like Excel, SharePoint, and Teams. It helps in the workflow of the projects.
- APIs for integration. This feature provides developers with sample code and APIs for embedding the Power BI dashboard in other software products.

Power BI Architecture

The whole process of data sourcing to the creation of reports and dashboards consists of four basic steps. Many programs and processes work together to get the required results.

Power BI Architecture

- Sourcing Data ?Power BI has a large range of data sources. Data can be imported from cloud-based online sources, and files in your system. There is a limit of 1 GB on importing data from online services. Power BI data sources are Excel, Text/CSV, XML, JSON, Oracle Database, Azure SQL Database, etc.
- Transforming the Data ?Cleaning and pre-processing are done before visualizing the data. It means removing missing values and useless data from the rows and columns. There are some rules to transforming and loading the datasets into the warehouse.

- Report and Publish ?Now, the report is created based on requirements, after cleaning and transforming the data. A report is data visualization with different filters and constraints represented in the form of graphs, pie, charts, etc.
- Creating Dashboards ?Power BI dashboards are created by pinning independent elements of live reports. It is created after the publishing of the report to the BI service. After reports get saved, the visual maintains the filter settings.

Power BI Architecture Components

- Power Query:Power Query is included in the Power Query Editor of Power BI Desktop, and It allows users to connect distinct information from multiple sources and convert them to satisfy their business requirements
- Power BI service:Power BI Service connects Power View, Power Pivot, Power BI Report Server, Power Q&A, and other components with the Workspace and allows you to connect with the data.
- Power Pivot:Power Pivot is a data modeling technique that uses the Data Analysis Expression (DAX) language to create data models.
- Power View:Power View creates graphs, maps, charts, and other visuals with drag-and-drop features. Power View can connect and filter different data sources to make a report on a single device.
- Power Map:Power Map allows you to base on the country, longitude, and latitude, Bing Maps shows the exact geospatial visuals of complex business information.
- Power BI Desktop:Power BI Desktop brings everything on a single platform, including Power Pivot, Power Query, Power View, etc. It is free software that you can install on a PC or laptop to create dashboards and reports.
- Power BI App:Power BI mobile applications are available for iOS, Android devices, and Windows. It does not matter whether your data is stored on the cloud or on-premises servers, you can access reports, run queries, share reports, and also get personalized notifications.
- Power Q&A:Power Q&A uses Natural Language Processing and integrates it with Cortana

(Microsoft's virtual assistant) to get the answers to your query.

Applications of Power BI

Microsoft Power BI is a powerful business intelligence tool that allows users to analyze data and share insights in the form of graphical representations. Some common practical applications of Power BI are:

Applications of Power BI

- Data Visualization
- Data Management
- Data Analytics With Internal Software Systems
- Business Reporting
- Provide complex data within software and apps
- Streamline Organizational Processes
- Data Collaboration and Sharing
- Enhance the marketing
- Real-time look at the company's financial performance with financial Dashboards
- Create Consistent Reporting Standards

Top Companies Using Power BI

Power BI is a popular business intelligence tool and is used by many companies. Here are some of the top companies that are using Microsoft Power BI for their business intelligence reports:

Top Companies Using Power BI

- Nestle
- Dell
- Vale
- Infosys Ltd
- Rolls-Royce Holdings PLC

- The North Face
- RHI Magnesita
- Cargill
- NielsenIQ
- L'Oréal
- Hospital Montfort

Top Companies Hiring For Power BI

Since Power BI is a popular tool among big and small businesses, many companies hire Power BI professionals. Here are some of the top companies that hire Power BI professionals:

Tech:

- Microsoft
- Amazon
- Google
- Salesforce
- Netflix

Finance:

- JPMorgan Chase
- Bank of America
- Citigroup
- American Express
- Goldman Sachs

Consulting:

- Accenture
- Deloitte
- EY

- KPMG

- PwC

Others:

- Starbucks

- Nike

- Tesla

- Pfizer

- Walmart

Others:

Prepare for Job interviews by visiting? [Top 30 Power BI Interview Questions and Answers](#)

Power BI Jobs and Salaries

Here is the list of job positions and salaries for Power BI:

Power BI Jobs and Salaries

Job Role | Average Salary (USD)

Power BI Analyst | \$75,000

Power BI Developer | \$90,000

Power BI Business Analyst | \$85,000

Power BI Architect | \$120,000

Power BI Data Analyst | \$80,000

Power BI Report Developer | \$70,000

Can I learn Power BI by myself

Yes, with the help of this Power BI tutorial anyone can learn Power BI by themselves at the comfort of their homes.

What PowerBI is used for?

It is used to gather, analyze, and transform data from various sources into interactive dashboards and visualizations that help businesses make better data-driven decisions.

What is the best way to learn PowerBI?

Here is the best way to learn PowerBI ?

What is the best way to learn PowerBI?

- Familiar with Basics like:- MsExcel etc
- Take Some Courses from GeeksforGeeks, etc
- Practising with real data on PowerBI
- Join the community on Different Platforms like:- LinkedIn etc.
- Stay up to date with new features.

How long does it take to learn PowerBI?

If you are a beginner and you have some fundamental knowledge of MS Excel then it will take around 4-6 weeks to learn PowerBI.

How Much that PowerBI Cost?

Microsoft PowerBI has three Different pricing plans ranging from INR 0 to INR 1570 you can choose from them depending upon your requirement.

How Much that PowerBI Cost?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

How Much that PowerBI Cost?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

How Much that PowerBI Cost?

S

How Much that PowerBI Cost?

- AI-ML-DS
- Power BI
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner

- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL

- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development

- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements

- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- What is Tableau and its Importance in Data Visualization?
- Tableau Installation Guide

- Tableau Drivers - Download and Connect
- Tableau - Data Terminology
- Power BI vs Tableau - Which one is Right for You?
- Difference Between Google Data Studio vs Tableau

Tableau Basics

- Tableau - Navigation
- Start Page in Tableau
- Tableau - Design Flow
- Tableau - File System
- Tableau - Data Types
- How to change Data type in Tableau?
- Show Me in Tableau

Working with Data Sources

- Tableau - Connect to a data source
- Import Excel file in Tableau
- Data Cleaning in Tableau
- Tableau - Join databases
- Tableau - Joining data files with inconsistent labels
- Data Blending in Tableau
- Replacing data source in Tableau
- Tableau - Split the text to columns
- Tableau - Displaying data in Worksheet
- Tableau - Adding, Renaming and duplicating worksheets

Visualization using Tableau

- Tableau - Create Visualization

- Tableau - Change the order in visualisation
- Tableau - Change the summary

Tableau Calculations

- Tableau - Operators
- IF Function in Tableau
- Aggregate Functions in Tableau
- Create Calculated Field in Tableau
- Logical Functions in Tableau
- Case Function in Tableau
- Highlight Tables in Tableau
- Quick Table Calculation in Tableau
- LOD expression in Tableau

Sort and Filter with Tableau

- Filtering in Visualization in Tableau
- Quick Filter in Tableau
- Context Filter in Tableau
- Conditional Filter in Tableau
- Slicing Filter in Tableau
- Grouping in Visualization in Tableau
- Hierarchy in Tableau
- Set in Tableau
- Combined Set in Tableau
- Parameters in Tableau
- Tableau Bins
- Manual Sorting of Visualization in Tableau
- Sorting by Data Source order in Visualization in Tableau

- Sorting by field in Visualization in Tableau

Different Charts in Tableau

- Line graph in Tableau
- Dual Lines Chart in Tableau
- Bar Chart in Tableau
- Side-by-Side Bars in Tableau
- Stacked Bars in Tableau
- Histogram in Tableau
- Cumulative Histogram in Tableau
- Scatter plot in Tableau
- Bubble Chart in Tableau
- Pie chart in tableau
- Crosstab in tableau
- Total in a crosstab by column in Tableau
- Box Plot in Tableau
- Bullet Graph in Tableau
- Area Chart in Tableau
- Heat Map in Tableau
- Basic Map in Tableau
- Geographical plot in Tableau
- Lollipop Chart in Tableau
- Circle views in Tableau
- Dual Combination Chart in Tableau
- Treemap in Tableau
- Funnel Chart in Tableau
- Traditional Funnel Chart in Tableau

- Waterfall Chart in Tableau
- Gantt Chart in Tableau

Working with Dashboards

- Create a Dashboard in Tableau
- Text Object on Dashboard in Tableau
- Image Object on Dashboard in Tableau
- Tableau - Objects on Dashboard
- Tableau - Filters in Dashboard
- Tableau - Device Preview
- Format Dashboard Layout in Tableau

Advance Tableau Topics

- Forecast in Tableau
- Reference line in Tableau
- Reference Band in Tableau
- Reference Distribution Band in Tableau
- Trend Lines in Tableau
- Tableau - Building up a storyline
- Word Cloud in Tableau

Tableau Tutorial

-
-
-

Tableau Tutorial

In this Tableau tutorial, we will learn about Tableau from basics to advance using the huge dataset containing topics like Tableau basics, working with different data sources, different charts available

in Tableau, etc. Tableau is a powerful tool used for data analysis and visualization. It allows the creation of amazing and interactive visualization and that too without coding. It provides the features like cleaning, organizing, and visualizing data.

Tableau Tutorial

Tableau is very famous as it can take in data and produce the required data visualization output in a very short time. Basically, it can elevate your data into insights that can be used to drive your action in the future. And Tableau can do all this while providing the highest level of security with a guarantee to handle security issues as soon as they arise or are found by users.

Tableau Tutorial

What is Tableau?

Tableau is a data visualization tool and it allows connecting with a large range of data sources, creating interactive visualizations, and providing features to share work with other team members. It is also used by data analysts and data scientists to explore data and create visualizations that communicate understandings to others. It has an interface that allows drag-and-drop data areas to create charts, graphs, and visualizations to analyze the data which is easier to use for beginners also. It is used by businesses like medicine, technology, e-commerce, etc, to analyze data and make data-driven judgments.

What is Tableau?

Once you've installed Tableau, you'll be prompted to activate your license or sign in with your Tableau account.

Connecting Data Sources

- To create a visualization in Tableau, we need to connect with the data source, which includes spreadsheets, cloud services, and databases.
- To connect to a data source, need to provide Tableau with the necessary information, login

certifications, and the place of the data source. after we connect to a data source, we are able to see the available data fields and start creating visualizations in Tableau for the analysis of data.

Table of Content ? Tableau Tutorial

- Introduction
- Tableau Basics
- Working with Data Sources
- Visualization using Tableau
- Tableau Calculations
- Sort and Filter with Tableau
- Different Charts in Tableau
- Working with Dashboards
- Advance Tableau Topics

Introduction

- What is Tableau and its Importance in Data Visualization?
- Tableau Installation Guide
- Tableau Drivers ? Download and Connect
- Tableau ? Data Terminology
- Power BI vs Tableau ? Top 5 Key Differences That You Should Know
- Difference Between Google Data Studio vs Tableau

Tableau Basics

- Tableau ? Navigation
- Start Page in Tableau
- Tableau ? Design Flow
- Tableau ? File System
- Tableau ? Data Types

- How to change Data type in Tableau?
- Show Me in Tableau

Working with Data Sources

- Tableau ? Connect to a data source
- Import Excel file in Tableau
- Data Cleaning in Tableau
- Tableau ? Join databases
- Joining data files with inconsistent labels
- Data Blending
- Replacing Data source
- Tableau ? Split the text to columns
- Tableau ? Displaying data in Worksheet
- Tableau ? Adding, Renaming and duplicating worksheets

Visualization using Tableau

- Create Visualization
- Change the order in visualization
- Change the summary

Tableau Calculations

- Tableau ? Operators
- IF Function in Tableau
- Aggregate Functions in Tableau
- Create Calculated Field in Tableau
- Logical Functions in Tableau
- Case Function in Tableau
- Highlight Tables in Tableau

- Quick Table Calculation in Tableau
- LOD Calculation

Sort and Filter with Tableau

- Filtering in Visualization in Tableau
- Quick Filter in Tableau
- Context Filter in Tableau
- Conditional Filter in Tableau
- Slicing Filter in Tableau
- Grouping in Visualization in Tableau
- Hierarchy in Tableau
- Set in Tableau
- Combined Set in Tableau
- Parameters in Tableau
- Tableau Bins
- Manual Sorting of Visualization in Tableau
- Sorting by Data Source order in Visualization in Tableau
- Sorting by field in Visualization in Tableau

Different Charts in Tableau

- Line graph in Tableau
- Dual Lines Chart in Tableau
- Bar Chart in Tableau
- Side-by-Side Bars in Tableau
- Stacked Bars in Tableau
- Histogram in Tableau
- Cumulative Histogram in Tableau
- Scatter plot in Tableau

- Bubble Chart in Tableau
- Pie chart in tableau
- Crosstab in tableau
- Total in a crosstab by column in Tableau
- Box Plot in Tableau
- Bullet Graph in Tableau
- Area Chart in Tableau
- Heat Map in Tableau
- Basic Map in Tableau
- Geographical plot in Tableau
- Lollipop Chart in Tableau
- Circle views in Tableau
- Dual Combination Chart in Tableau
- Treemap in Tableau
- Funnel Chart in Tableau
- Traditional Funnel Chart in Tableau
- Waterfall Chart in Tableau
- Gantt Chart in Tableau

Working with Dashboards

- Create a Dashboard in Tableau
- Text Object on Dashboard in Tableau
- Image Object on Dashboard in Tableau
- Tableau ? Objects on Dashboard
- Tableau ? Filters in Dashboard
- Tableau ? Device Preview
- Format Dashboard Layout in Tableau

Advance Tableau Topics

- Forecast in Tableau
- Reference line in Tableau
- Reference Band in Tableau
- Reference Distribution Band in Tableau
- Trend Lines in Tableau
- Tableau ? Building up a storyline
- Word Cloud in Tableau

Q.1 What is Tableau?

Answer: Tableau is a data visualization tool that allows you to analyze and display data in an intuitive and user-friendly way.

Q.2 Is Tableau difficult to learn?

Answer: While Tableau has a learning curve, it's generally considered to be one of the more user-friendly data visualization tools available.

Q.3 Can I use Tableau for free?

Answer: Yes, you can use Tableau Public for free. Tableau Public allows you to create and share visualizations online, but your data will be publicly accessible.

Q.4 What kind of data sources can I connect to with Tableau?

Answer: Tableau can connect to a wide range of data sources, including spreadsheets, databases, and cloud services.

Q.5 Is Tableau suitable for small businesses?

Answer: Yes, Tableau is suitable for businesses of all sizes. Tableau offers pricing plans that are affordable for small businesses and startups.

Q.5 Is Tableau suitable for small businesses?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Q.5 Is Tableau suitable for small businesses?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Q.5 Is Tableau suitable for small businesses?

- AI-ML-DS
- Tableau

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest

- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- NumPy Introduction
- Python Numpy
- NumPy Array in Python
- Basics of NumPy Arrays
- Numpy - ndarray
- Data type Object (dtype) in NumPy Python

Creating NumPy Array

- Numpy - Array Creation
- numpy.arange() in Python
- numpy.zeros() in Python
- NumPy - Create array filled with all ones
- NumPy - linspace() Function
- numpy.eye() in Python
- Creating a one-dimensional NumPy array
- How to create an empty and a full NumPy array?
- Create a Numpy array filled with all zeros - Python
- How to generate 2-D Gaussian array using NumPy?
- How to create a vector in Python using NumPy
- Python - Numpy fromrecords() method

NumPy Array Manipulation

- NumPy Copy and View of Array
- How to Copy NumPy array into another array?
- Appending values at the end of an NumPy array
- How to swap columns of a given NumPy array?
- Insert a new axis within a NumPy array
- numpy.hstack() in Python

- `numpy.vstack()` in python
- Joining NumPy Array
- Combining a one and a two-dimensional NumPy Array
- Python | Numpy `np.ma.concatenate()` method
- Python | Numpy `dstack()` method
- Splitting Arrays in NumPy
- How to compare two NumPy arrays?
- Find the union of two NumPy arrays
- Find unique rows in a NumPy array
- Python | Numpy `np.unique()` method
- `numpy.trim_zeros()` in Python

Matrix in NumPy

- Matrix manipulation in Python
- numpy matrix operations | `empty()` function
- numpy matrix operations | `zeros()` function
- numpy matrix operations | `ones()` function
- numpy matrix operations | `eye()` function
- numpy matrix operations | `identity()` function
- Adding and Subtracting Matrices in Python
- Matrix Multiplication in NumPy
- Numpy `ndarray.dot()` function | Python
- NumPy | Vector Multiplication
- How to calculate dot product of two vectors in Python?
- Multiplication of two Matrices in Single line using Numpy in Python
- Python | Numpy `np.eigvals()` method
- How to Calculate the determinant of a matrix using NumPy?

- Python | Numpy matrix.transpose()
- Python | Numpy matrix.var()
- Compute the inverse of a matrix using NumPy

Operations on NumPy Array

- Numpy | Binary Operations
- Numpy | Mathematical Function
- Numpy - String Functions & Operations

Reshaping NumPy Array

- Reshape NumPy Array
- Python | Numpy matrix.resize()
- Python | Numpy matrix.reshape()
- NumPy Array Shape
- Change the dimension of a NumPy array
- numpy.ndarray.resize() function - Python
- Flatten a Matrix in Python using NumPy
- numpy.moveaxis() function | Python
- numpy.swapaxes() function | Python
- Python | Numpy matrix.swapaxes()
- numpy.vsplit() function | Python
- numpy.hsplit() function | Python
- Numpy MaskedArray.reshape() function | Python
- Python | Numpy matrix.squeeze()

Indexing NumPy Array

- Basic Slicing and Advanced Indexing in NumPy
- numpy.compress() in Python

- Accessing Data Along Multiple Dimensions Arrays in Python Numpy
- How to access different rows of a multidimensional NumPy array?
- `numpy.tril_indices()` function | Python

Arithmetic operations on NumPyArray

- NumPy Array Broadcasting
- Estimation of Variable | set 1
- Python: Operations on Numpy Arrays
- How to use the NumPy sum function?
- `numpy.divide()` in Python
- `numpy.inner()` in python
- Absolute Deviation and Absolute Mean Deviation using NumPy | Python
- Calculate standard deviation of a Matrix in Python
- `numpy.gcd()` in Python

Linear Algebra in NumPy Array

- Numpy | Linear Algebra
- Get the QR factorization of a given NumPy array
- How to get the magnitude of a vector in NumPy?
- How to compute the eigenvalues and right eigenvectors of a given square array using NumPY?

NumPy and Random Data

- Random sampling in numpy | `randf()` function
- Random sampling in numpy | `random()` function
- Random sampling in numpy | `random_sample()` function
- Random sampling in numpy | `sample()` function
- Random sampling in numpy | `random_integers()` function
- Random sampling in numpy | `randint()` function

- `numpy.random.choice()` in Python
- How to choose elements from the list with different probability using NumPy?
- How to get weighted random choice in Python?
- `numpy.random.shuffle()` in python
- `numpy.random.geometric()` in Python
- `numpy.random.permutation()` in Python

Sorting and Searching in NumPy Array

- Searching in a NumPy array
- NumPy Array Sorting | How to sort NumPy Array
- Numpy | Sorting, Searching and Counting
- Variations in different Sorting techniques in Python
- `numpy.sort_complex()` in Python
- Python | Numpy `np.ma.mini()` method
- Python | Numpy matrix.`sort()`
- Python | Numpy matrix.`argsort()`

Universal Functions

- NumPy ufuncs | Universal functions
- Create your own universal function in NumPy

Working With Images

- Create a white image using NumPy in Python
- Convert a NumPy array to an image
- How to Convert images to NumPy array?
- How to Convert an image to NumPy array and save it to CSV file using Python?

Projects and Applications with NumPy

- Python program to print checkerboard pattern of nxn using numpy

- Implementation of neural network from scratch using NumPy
- Analyzing selling price of used cars using Python

NumPy Tutorial ? Python Library

-
-
-

NumPy Tutorial ? Python Library

NumPy is a powerful library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy's array objects are more memory-efficient and perform better than Python lists, which is essential for tasks in scientific computing, data analysis, and machine learning. This NumPy tutorial will cover core features, and all concepts from basic to advanced divided in 10 sections.

Section 1: NumPy Arrays

The NumPy array, also called `ndarray`, is a grid of values, all of the same type. They can be one-dimensional (like a list), two-dimensional (like a matrix), or multi-dimensional (like a table with rows and columns).

Section 1: NumPy Arrays

To understand all the basics of NumPy Arrays ? explaining their types (one-dimensional and multi-dimensional), key attributes (axis, shape, rank, dtype): Basics of NumPy Arrays

Section 2: Creating Arrays in Numpy

NumPy arrays are created using the `np.array()` function, which converts lists, tuples, or other sequences into a NumPy array. You can create different types of arrays, such as 1D arrays from a simple list of elements, 2D arrays from nested lists representing rows and columns,

and multi-dimensional arrays by further nesting lists.

Section 2: Creating Arrays in Numpy

Example of a simple one-dimensional array:

Section 2: Creating Arrays in Numpy

To understand all ways for creating Array refer to: [Numpy array creation page](#). For understanding techniques in-depth (explained in numpy creation page) refer to following resources:

Section 2: Creating Arrays in Numpy

- arange Method? Creates an array with evenly spaced values within a specified range.
- zero Method? Creates an array filled with zeros.
- array filled with all ones
- linspace Method? array of evenly spaced values over a specified interval.
- eye Method? 2D array with ones on the diagonal and zeros elsewhere (identity matrix)
- create an empty and a full NumPy array

Section 2: Creating Arrays in Numpy

Note: These numPy array creation methods form the foundation for efficiently working with arrays in data science, machine learning, and scientific computing.

Section 4: Operations in Numpy Array

NumPy arrays offer four essential types of operations that allow efficient data manipulation by performing element-wise computations, mathematical functions, string processing, and logical comparisons.

Section 4: Operations in Numpy Array

- Binary Operations Perform element-wise comparisons (e.g., greater than, equal to) and logical operations like AND or OR.
- Mathematical Operations? Functions such as `sum()`, `mean()`, and `sqrt()` allow for quick and efficient

numerical calculations.

- String Operations? Use `np.char()` functions for operations like concatenation, uppercasing, and replacing strings within arrays.
- Arithmetic Operations? Enable fast element-wise addition, subtraction, multiplication, and division in Arrays.

Section 5: Slicing & Indexing in Numpy Array

While indexing is used to access specific elements in an array based on their positions or conditions, slicing is used to extract a subset of elements from an array using the syntax `array[start:stop:step]`. It works for both 1D and multi-dimensional arrays, making it easy to select specific rows, columns, or ranges of data.

Section 5: Slicing & Indexing in Numpy Array

- Slicing and Indexing
- Get hands on practice with more example of Array Indexing
- Slicing Multidimensional Array for accessing different rows

Section 6: Shaping & Reshaping in Array

Shape of an array can be defined as the number of elements in each dimension. It can be accessed using the `shape` attribute, which returns a tuple representing the dimensions of the array. In this section, we will explore how to change the shape of a NumPy array. This includes reshaping, flattening, and modifying the structure of arrays to suit specific tasks.

Section 6: Shaping & Reshaping in Array

- For shape manipulation - NumPy array
- Array reshaping
- Flattening of numpy array- changing the order

Section 7: Sorting and Searching in Array

Sorting in NumPy refers to arranging the elements of an array in a specific order, either ascending or descending. The `np.sort()` function is used for this purpose. By default, it sorts elements in ascending order, and descending order can be achieved using slicing techniques like `[::-1]`.

Section 7: Sorting and Searching in Array

Note: Sorting works for both single-dimensional and multi-dimensional arrays, click on links belows for in-depth understanding:

Section 7: Sorting and Searching in Array

- Sorting NumPy Array
- Different types of Sorting Technique

Section 7: Sorting and Searching in Array

Searching in NumPy involves finding specific values or conditions within an array. In this section, we'll explore different techniques for searching within NumPy arrays searching for Specific Values using `np.where()` , `np.searchsorted()` and `np.nonzero()` that returns the indices of all non-zero elements.

Section 7: Sorting and Searching in Array

- Dive deeper with Searching in Numpy Array
- Numpy Array ? Searching , Sorting , Counting

Section 8: Combining, Splitting, and Aggregating Arrays

Combining arrays involves merging smaller arrays into a single larger one. arrays can be combined vertically, horizontally, or along any specific axis. It includes Techniques like concatenation and stacking (horizontal, vertical).

Section 8: Combining, Splitting, and Aggregating Arrays

- Joining Numpy Array
- Combining 1-D and 2-D array

Section 8:Combining, Splitting, and Aggregating Arrays

Splitting arrays is the process of dividing a larger array into smaller, manageable sub-arrays. The division can occur along rows, columns, or other axes.

Section 8:Combining, Splitting, and Aggregating Arrays

Splitting arrays in Numpy

Section 8:Combining, Splitting, and Aggregating Arrays

Aggregation refers to summarizing data within an array by applying mathematical operations like summing, finding the average, or determining the maximum/minimum values.

Section 8:Combining, Splitting, and Aggregating Arrays

- Summing Function- In this we use `numpy.sum()` to do the sum of array elements.
- Average Function ? In this we use `numpy.mean()` to find out the average of array elements.

Section 9: Matrix Operations

In NumPy, a matrix is represented as a 2D array. Matrix operations are a fundamental part of linear algebra. Matrix Addition, Subtraction, and Multiplication are fundamental for manipulating matrices. To work with matrices, NumPy provides simple tools. For example, `np.transpose()` flips the matrix by turning rows into columns and columns into rows. If you want to change the shape of a matrix, like turning a single row into multiple rows, you use `np.reshape()`. To simplify a matrix and turn it into a single list of values, you can use `np.flatten()`. Finally, `np.dot()` is used for multiplying two matrices.

Section 9: Matrix Operations

- To learn more about the Matrix Operations in Numpy: Matrix Addition, Subtraction, and Multiplication. Now, there are many more matrix operations like:
- Matrix Transpose
- Determinant and Inverse of a Matrix
- Matrix Operation ? calculating dot product of two vectors

- Find variance of a matrix

Section 10: Linear Algebra in Numpy Array

Eigenvalues and eigenvectors are fundamental concepts in linear algebra. NumPy provides a robust `numpy.linalg` module to perform various linear algebra operations efficiently.

Section 10: Linear Algebra in Numpy Array

- Numpy Array Linear Algebra
- QR factorization of the Numpy Array
- Computing the eigen values

Section 11: Numpy and Random Data

NumPy provides a powerful module, `numpy.random`, for generating random data efficiently. It enables users to create random numbers, samples, and arrays for a variety of distributions.

Section 11: Numpy and Random Data

The module allows for generating random integers, random floats between 0 and 1, and random samples from normal, uniform, and other statistical distributions. For example: if you want to generate 5 random numbers from the normal distribution using NumPy. Click on the link for solution.

Section 12: Advance Numpy

Welcome to the advanced section of NumPy! If you've already learned the basics of arrays, you're ready to explore some powerful tools that make working with data faster and easier.

Section 12: Advance Numpy

- Numpy Array Broadcasting? Broadcasting lets you do operations on arrays of different shapes without changing their size or shape.
- Numpy Array Vectorization? Instead of using slow loops, vectorization uses optimized functions to work on the whole array at once.
- Universal Function in Numpy? Ufuncs are built-in functions in NumPy that perform operations on

each element of an array. Example: `np.add()` for addition, `np.sqrt()` for square root, and `np.sin()` for sine values.

- Working With Images in numpy? In NumPy, you can treat images as multi-dimensional arrays (like 2D or 3D arrays).

Section 12: Advance Numpy

Refer to Practice Exercises, Questions, and Solutions for hands-on-numpy problems

Section 12: Advance Numpy

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the Three 90 Challenge? complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

Section 12: Advance Numpy

- AI-ML-DS
- Numpy
- Python
- Python-numpy
- Tutorials

Section 12: Advance Numpy

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy

- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- Pandas Introduction
- How to Install Pandas in Python?
- How To Use Jupyter Notebook - An Ultimate Guide

Creating Objects

- Creating a Pandas DataFrame
- Python Pandas Series
- Creating a Pandas Series

Viewing Data

- Python | Pandas Dataframe/Series.head() method
- Python | Pandas Dataframe/Series.tail() method
- Pandas DataFrame describe() Method

Selection & Slicing

- Dealing with Rows and Columns in Pandas DataFrame
- Python | Pandas Extracting rows using .loc[]

- Extracting rows using Pandas .iloc[] in Python
- Indexing and Selecting Data with Pandas
- Boolean Indexing in Pandas
- Python | Pandas DataFrame.ix[]
- Python | Pandas Series.str.slice()
- How to take column-slices of DataFrame in Pandas?

Operations

- Python | Pandas.apply()
- Apply function to every row in a Pandas DataFrame
- Python | Pandas Series.apply()
- Python | Pandas dataframe.aggregate()
- Pandas DataFrame mean() Method
- Python | Pandas Series.mean()
- Python | Pandas dataframe.mad()
- Python | Pandas Series.mad() to calculate Mean Absolute Deviation of a Series
- Python | Pandas dataframe.sem()
- Python | Pandas Series.value_counts()
- Python | Pandas Index.value_counts()
- Applying Lambda functions to Pandas Dataframe

Manipulating Data

- Adding New Column to Existing DataFrame in Pandas
- Python | Delete rows/columns from DataFrame using Pandas.drop()
- Python | Pandas DataFrame.truncate
- Python | Pandas Series.truncate()
- Iterating over rows and columns in Pandas DataFrame
- Pandas Dataframe.sort_values()

- Python | Pandas Dataframe.sort_values() | Set-2
- How to add one row in existing Pandas DataFrame?

Grouping Data

- Pandas GroupBy
- Grouping Rows in pandas
- Combining Multiple Columns in Pandas groupby with Dictionary

Merging, Joining, Concatenating and Comparing

- Python | Pandas Merging, Joining, and Concatenating
- Python | Pandas Series.str.cat() to concatenate string
- Python - Pandas dataframe.append()
- Python | Pandas Series.append()
- Python | Pandas Index.append()
- Python | Pandas Series.combine()
- Add a row at top in pandas DataFrame
- Python | Pandas str.join() to join string/list elements with passed delimiter
- Join two text columns into a single column in Pandas
- How To Compare Two Dataframes with Pandas compare?
- How to compare the elements of the two Pandas Series?

Working with Date and Time

- Python | Working with date and time using Pandas
- Python | Pandas Timestamp.timestamp
- Python | Pandas Timestamp.now
- Python | Pandas Timestamp.isoformat
- Python | Pandas Timestamp.date
- Python | Pandas Timestamp.replace

- Python | Pandas.to_datetime()
- Python | pandas.date_range() method

Working With Text Data

- Python | Pandas Working With Text Data
- Python | Pandas Series.str.lower(), upper() and title()
- Python | Pandas Series.str.replace() to replace text in a series
- Python | Pandas Series.replace()
- Python | Pandas Series.str.strip(), lstrip() and rstrip()
- Python | Pandas tseries.offsets.DateOffset

Working with CSV and Excel files

- Pandas Read CSV in Python
- Saving a Pandas Dataframe as a CSV
- Loading Excel spreadsheet as pandas DataFrame
- Creating a dataframe using Excel files
- Python | Working with Pandas and XlsxWriter | Set - 1
- Python | Working with Pandas and XlsxWriter | Set ? 2
- Python | Working with Pandas and XlsxWriter | Set ? 3

Visualization

- Data Visualization with Pandas
- Data analysis and Visualization with Python
- Data Analysis and Visualization with Python | Set 2
- Box plot visualization with Pandas and Seaborn

Applications and Projects

- How to Do a vLookup in Python using pandas
- Convert CSV to HTML Table in Python

- KDE Plot Visualization with Pandas and Seaborn
- Analyzing selling price of used cars using Python
- Add CSS to the Jupyter Notebook using Pandas

Pandas Tutorial

-
-
-

Pandas Tutorial

Pandas is a powerful data manipulation and analysis library for Python. It provides data structures like series and dataframes to effectively and easily clean, transform, and analyze large datasets and integrates seamlessly with other Python libraries, such as NumPy and matplotlib. It offers powerful functions for data transformation, aggregation, and visualization, which are crucial for effective analysis.

Pandas Tutorial

Pandas revolves around two primary data structures: series (1D) for single columns and dataframe (2D) for tabular data, enabling efficient data manipulation.

Section 1: Installing and Importing Pandas

To install Pandas, run the following command:

Section 1: Installing and Importing Pandas

```
pip install pandas
```

Section 1: Installing and Importing Pandas

To use Pandas in your code, import it with:

Section 1: Installing and Importing Pandas

```
import pandas as pd
```

Section 1: Installing and Importing Pandas

This imports the Pandas library and gives it the alias `pd` for convenience.

Section 2: Creating DataFrames

- DataFrame is a two-dimensional table-like data structure with labeled rows and columns, where each column can have a different data type (e.g., integers, strings, floats). It can be created from Python data structures like lists, dictionaries, or a list of dictionaries.

Section 2: Creating DataFrames

Example: Create DataFrame using dictionary:

Section 2: Creating DataFrames

In this example, a dictionary named `data` is created with keys representing column names (Name, Age) and values as lists containing the respective data. `pd.DataFrame()` function is then used to convert this dictionary into a DataFrame, which is stored in the variable `df`.

Section 2: Creating DataFrames

To learn Different ways to create dataframe, refer: [Creating a Pandas DataFrame page](#).

Section 2: Creating DataFrames

- Series in Pandas is 1-dimensional labeled array capable of holding any data type (integers, strings, floats, etc.). Each element is associated with an index, either default (0, 1, 2?) or custom labels. It can be created from lists, NumPy arrays, dictionaries, or scalar values.

Section 2: Creating DataFrames

Example: Creating a Series

Section 2: Creating DataFrames

To Learn more ways to creating pandas series, refer: [Creating a Pandas Series](#).

Section 3 : Reading CSV Files

CSV (Comma Separated Values)files are a common format for storing large datasets in plain text. The Pandas library in Python provides,read_csv() function, to load these files into a DataFrame. For our example we will usepeople.csv.

Section 3 : Reading CSV Files

Output:

Section 3 : Reading CSV Files

Pandas Read CSV

Section 3 : Reading CSV Files

Note:With this line, pandas will display only the first and last 5 rows if the DataFrame is large.

Section 3 : Reading CSV Files

- Usingto_string()prints the entire DataFrame. Pandas has options to control how much of the DataFrame is displayed.
- You can check or set these using:pd.options.display.max_rows.

Section 3 : Reading CSV Files

Pandas provides flexible options to work with various file formats, including CSV andJSON (JavaScript Object Notation). Using read_json() function in pandas, we can read JSON files. To load a JSON file into a Pandas DataFrame, use the following code:

Section 3 : Reading CSV Files

Note: To run the code and learn about how theread_json()function handles different JSON structures with various parameters, refer to the page:Reading JSON Files.

Section 4:Understanding and Analyzing the Data Frame

After creating or loading a DataFrame, inspecting and summarizing the data is an important step in

understanding dataset. Pandas provides various functions to help you view and analyze the data.

Section 4:Understanding and Analyzing the Data Frame

- `head()`:View the first n rows of the DataFrame (default is 5 rows).
- `tail()`:View the last n rows of the DataFrame (default is 5 rows).
- `info()`:This method provides a concise summary of the DataFrame, including the number of non-null entries, column names, and data types. Let's see an example demonstrating the use of `head()`, `tail()`, and `info()` methods:

Section 4:Understanding and Analyzing the Data Frame

Output:

Section 4:Understanding and Analyzing the Data Frame

Viewing and Inspecting Data With Pandas

Section 4:Understanding and Analyzing the Data Frame

Pandas offers versatile tools for viewing and inspecting data, enabling to quickly summarize or dive deeper into specific data attributes. To explore more advanced functionalities in Pandas, you can follow the links below:

Section 4:Understanding and Analyzing the Data Frame

- View basic statistical details
- Get information about the DataFrame (data types, non-null counts, etc.)
- Check for missing data
- Inspect the DataFrame shape (rows and columns)
- Checking rows with minimum and maximum values

Section 4:Understanding and Analyzing the Data Frame

Note:`describe`generates descriptive statistics for numerical columns (such as count, mean, standard deviation, etc.). You can apply it to the entire DataFrame or specific columns. (You can also use

.describe(include='all')to include non-numeric data in the summary.)

Section 5: Indexing in Pandas

Indexing in Pandas refers to process of accessing and selecting data from a Pandas DataFrame or Series. There are multiple ways to do this. We will cover how to basic indexing, select specific columns , apply slicing, and use Boolean indexing to filter data efficiently.

Section 5: Indexing in Pandas

Example :Basic Indexing(Selecting a single column) with use of [] operator:

Section 5: Indexing in Pandas

We can also select multiple columns by passing a list of column names. For example, you can select both the 'Name' and 'Age' columns by providing a list ['Name', 'Age']. When it comes to selecting rows: we can use .loc[] to select rows by label, meaning we refer to the row index or label directly. Alternatively, .iloc[] allows for position-based indexing, where we select rows by their integer positions.

Section 5: Indexing in Pandas

For more examples where we have covered selecting single and multiple columns and row based selection: Indexing and Selecting Data with Pandas. Explore the following resources for advanced indexing techniques:

Section 5: Indexing in Pandas

- Using loc[] and iloc[] for Advanced Row and Column Selection
- Boolean Indexing in Pandas
- Slicing dataFrames and series: Part of indexing in Pandas, refers to selecting a subset of rows or columns from a DataFrame or Series by specifying a range of indices.

Section 6: Selecting and Filtering Data ? Techniques to Filter Based on Conditions

For selecting and filtering the data after indexing: is about narrowing down the data by applying

conditions to select only those rows or columns that meet certain criteria.

Section 6: Selecting and Filtering Data ? Techniques to Filter Based on Conditions

In this example, we filter the rows where the Age column is greater than 30. This is a simple condition that helps narrow down the data to only those rows that meet the condition. For more methods, refer to our page :Selecting Rows from a Dataframe based on Column Values. You can also refer to following resources for advanced techniques and more conditions in filtering and selection:

Section 6: Selecting and Filtering Data ? Techniques to Filter Based on Conditions

- Select Rows With Multiple Filters in Pandas
- Filtering by Column Value
- Filter rows based on a list of values
- Filter Pandas Dataframe with multiple conditions

Section 7: Dealing with Rows and Columns in Pandas DataFrame

Involves various data manipulation techniques in Pandas including adding and deleting columns, truncating data, iterating over DataFrames, and sorting data. Dealing with Rows and Columns in Pandas DataFrame? every concept and step is explained in this post.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

1. Adding a New Column to DataFrame: There are several methods available, each suitable for specific use cases. We can easily add new columns by assigning values to them by direct assignment.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

There are multiple methods, for that refer to: Adding new column to existing dataframe in pandas. If you want to add columns from one dataframe to another, refer to Adding Columns from Another DataFrame.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

2. Renaming columns using the `rename()` method or by directly modifying the `columns` attribute.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

- Use `rename()` for selective renaming.
- Use `columns` attribute for renaming all columns at once.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

We can also: Rename column by index in Pandas, refer to the article for examples.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

3. Reindexing Data with Pandas: allows to change the row and column labels of a DataFrame or Series. This operation is useful for aligning data with a new set of labels, handling missing data, and reorganizing datasets. The `reindex()` method can change the row indices or column labels of a DataFrame.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

Note: If the new index includes labels not present in the original DataFrame, the corresponding values will be set to NaN by default.

Section 7: Dealing with Rows and Columns in Pandas DataFrame

What if you want to reset the index? for that refer to: Convert Index to Column in Pandas DataFrame

Section 8: Handling Missing Data in Pandas

Working with Missing Data in Pandas is one of the most common tasks in data manipulation. Pandas provides various functions to identify, fill, and drop missing values efficiently. To explore the procedure hand-to-hand in detail refer to below steps:

Section 8: Handling Missing Data in Pandas

1. Identifying Missing Data With Pandas using:

Section 8: Handling Missing Data in Pandas

- `isnull()`: Returns True where values are missing (NaN), and False where values are present.
- `notnull()`: Returns True where values are present, and False where they are missing.

Section 8: Handling Missing Data in Pandas

Here's an example:

Section 8: Handling Missing Data in Pandas

2. Filling Missing Data: To replace missing values with a specific value, `fillna()` is used. For example: you want to fill missing values with default value 0 you can perform it like `df.fillna(0)`.

Section 8: Handling Missing Data in Pandas

We can also fill missing values with below techniques:

Section 8: Handling Missing Data in Pandas

- Replacing with the mean, median, or mode of the column
- forward fill ? Fill missing values with the last valid observation
- backward fill- Fill missing values with the next valid observation

Section 8: Handling Missing Data in Pandas

3. Dropping Missing Data With Pandas: We can also remove rows or columns that contain missing values using `dropna()`. For example:

Section 8: Handling Missing Data in Pandas

We can also use `df.dropna(axis=1)` to remove columns or `df.dropna(subset=['Column1', 'Column2'])` to drop rows with missing values in specific columns. For understanding various ways for which we can use drop refer to:

Section 8: Handling Missing Data in Pandas

- Dropping rows from Pandas dataframe with missing values
- Dropping one or multiple columns

Section 9: Aggregation and Grouping With Pandas

Aggregation and grouping in Pandas are powerful tools for analyzing and summarizing data. Grouping allows to segment your data into categories, while aggregation performs operations (like sum, mean, or count) on these groups to derive insights. `groupby()` function is commonly used for grouping data, followed by aggregation methods like `sum()`, `mean()`, or custom functions for statistical analysis.

Section 9: Aggregation and Grouping With Pandas

Examples of operations:

Section 9: Aggregation and Grouping With Pandas

In the example above:

Section 9: Aggregation and Grouping With Pandas

- `groupby('Category')`: Groups data by the 'Category' column.
- `.sum()`, `.mean()`, and `.size()`: Perform aggregation on the grouped data.

Section 9: Aggregation and Grouping With Pandas

- Grouping and Aggregating with Pandas? this is the main page where in-depth explanations and examples are given.
- For more advanced operations, you can use `agg()` to apply custom aggregation functions.
- Grouping Rows in pandas

Section 10: Merging and Concatenating Data in Pandas

This section covers techniques for combining multiple DataFrames or Series into a single DataFrame. These operations are essential for integrating datasets and can be performed in several ways:

Section 10: Merging and Concatenating Data in Pandas

1. Merging DataFrames: Combines data based on common column or index using functions

like merge or join. There are 4 types of joins:

Section 10: Merging and Concatenating Data in Pandas

- Inner Join: Keep rows that match in both DataFrames.
- Left Join: Keep all rows from the left DataFrame and match data from the right.
- Right Join and Outer Join are similar but differ in data retention rules.

Section 10: Merging and Concatenating Data in Pandas

Refer : Different types of Joins in pandas

Section 10: Merging and Concatenating Data in Pandas

Example of operations:

Section 10: Merging and Concatenating Data in Pandas

For examples of all operations:

Section 10: Merging and Concatenating Data in Pandas

- Merging Dataframes using merge
- Joining the dataframes ? Join method

Section 10: Merging and Concatenating Data in Pandas

2. Concatenating Data: Refers to stacking DataFrames either vertically (adding rows) or horizontally (adding columns). This can be achieved using the `pd.concat()` function. Let's create two dataFrames and concatenate it with the original one:

Section 10: Merging and Concatenating Data in Pandas

In the code:

Section 10: Merging and Concatenating Data in Pandas

- The `axis=0` parameter specifies concatenation along rows (vertically).
- `ignore_index=True` resets the index so it doesn't retain the original indices from both DataFrames.

Section 10: Merging and Concatenating Data in Pandas

LearnConcatenate Two or More Pandas DataFrameswith all operations and multiple examples. For more, Go Through below articles:

Section 10: Merging and Concatenating Data in Pandas

- Concatenate values in different columns to one column
- Merge two dataFrames on certain columns
- Merge multiple CSV Files into single dataframe

Section 11: Reshaping Data in Pandas

Reshaping data involves changing the structure of rows and columns to better organize or analyze data.Common operations include:

Section 11: Reshaping Data in Pandas

1. Pivot Tables in Pandas: Reshape data based on column values. The.pivot_table()methodis particularly powerful for creating aggregated views of data.

Section 11: Reshaping Data in Pandas

For more examples and in-depth understanding:Pivoting tables in pandas

Section 11: Reshaping Data in Pandas

2. Melting and Unmelting: Multiple columns are combined into a single key-value pair.

Section 11: Reshaping Data in Pandas

When you melt data:

Section 11: Reshaping Data in Pandas

- The column names become thekeys.
- The values in the columns become thevalues.

Section 11: Reshaping Data in Pandas

Note: To melt data, you must specify columns that act as ?identifiers? (id_vars) and others that need to be melted (value_vars).

Section 11: Reshaping Data in Pandas

More examples with :Reshaping ? Melt and Unmelt

Section 11: Reshaping Data in Pandas

3. Stacking and Unstacking With Pandas: Reshape the data by changing its rows and columns, particularly when working with MultiIndex DataFrames. These techniques help pivot the layout of your data.

Section 11: Reshaping Data in Pandas

- Stacking: Moves the data from columns to rows (making the DataFrame taller).
- Unstacking: Moves the data from rows to columns (making the DataFrame wider). Let's take a simple DataFrame with multiple levels of columns.

Section 11: Reshaping Data in Pandas

Explanation:

Section 11: Reshaping Data in Pandas

- Columns are stacked into rows.
- Now, the data has a MultiIndex for rows: First level: Product (e.g., Product_A, Product_B) Second level: Category (e.g., Sales, Expenses).
- First level: Product (e.g., Product_A, Product_B)
- Second level: Category (e.g., Sales, Expenses).
- Q1 and Q2 remain as columns

Section 11: Reshaping Data in Pandas

- First level: Product (e.g., Product_A, Product_B)
- Second level: Category (e.g., Sales, Expenses).

Section 11: Reshaping Data in Pandas

Let's compare and understand in-depth with more examples :stack,unstack and melt method. We can also use transpose method forSwapping Rows and Columns

Section 12: Handling Large DatasetsWith Pandas

When working with large datasets or performing intensive computations,optimizing performance in Pandasis required. Below are some techniques to improve the efficiency of your data processing workflows.

Section 12: Handling Large DatasetsWith Pandas

- Working with Chunks of Data: When dealing with large datasets that may not fit into memory, you can use thechunksizeparameter inpd.read_csv()to load the data in chunks that allows to process the data in manageable piecesrather than loading everything at once.
- Optimizing Memory Usage by Changing Data Types: By default, Pandas may use larger data types than necessary (e.g.,int64instead ofint8), which can lead to inefficient memory usage. Changing the data types with.astype()can optimize memory consumption significantly, especially with large datasets.

Section 12: Handling Large DatasetsWith Pandas

For more methods, click on link.

Section 12: Handling Large DatasetsWith Pandas

Refer toPandas Exercises and Programsfor hands-on practice to reinforce your understanding of key concepts, including data manipulation, cleaning, and analysis.

What is pandas used for in python?

Python Pandas is used for data manipulation, analysis, and cleaning. It simplifies handling structured data like spreadsheets and SQL tables. With Pandas, you can import data, clean it, transform it, and perform operations such as grouping, merging, and aggregating.

Is Python pandas easy to learn?

Yes, Python Pandas is relatively easy to learn, especially for those with basic Python knowledge.

How to start with Python pandas?

To start with Pandas, you can follow an introduction to Pandas Python guide.

Is pandas harder than SQL?

Pandas and SQL serve different purposes, but neither is inherently harder. SQL is used for database queries, while Pandas provides more flexibility for in-memory data manipulation. If you're familiar with SQL queries, you might find Pandas query examples helpful to bridge the gap between SQL and Pandas.

What is the full form of Pandas?

The full form of Pandas is 'Python Data Analysis Library,' derived from the term 'panel data.' It's designed for data manipulation and analysis, making it indispensable in data science tasks.

What are Pandas best used for?

Pandas are best used for tasks like data cleaning, transformation, and analysis. You can work with time-series data, perform merging, and handle missing values with ease.

When to use pandas?

Use Pandas when working with structured datasets like spreadsheets, databases, or CSV files. It's ideal for data cleaning, transformation, analysis, and visualization.

When to use pandas?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

When to use pandas?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get

a 90% refund. What more motivation do you need? Start the challenge right away!

When to use pandas?

- AI-ML-DS
- Pandas
- Python-pandas

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner

- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL

- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development

- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements

- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- Matplotlib Tutorial
- Environment Setup for Matplotlib

- Using Matplotlib with Jupyter Notebook
- Introduction to Matplotlib
- Matplotlib Pyplot
- Matplotlib - Axes Class

Multiple Plots

- How to Create Multiple Subplots in Matplotlib in Python?
- How to Add Title to Subplots in Matplotlib?
- How to Set a Single Main Title for All the Subplots in Matplotlib?
- How to Turn Off the Axes for Subplots in Matplotlib?
- How to Create Different Subplot Sizes in Matplotlib?
- How to set the spacing between subplots in Matplotlib in Python?

Working with Legends

- Matplotlib.pyplot.legend() in Python
- Matplotlib.axes.Axes.legend() in Python
- Change the legend position in Matplotlib
- How to Change Legend Font Size in Matplotlib?
- How Change the vertical spacing between legend entries in Matplotlib?
- Use multiple columns in a Matplotlib legend
- How to Create a Single Legend for All Subplots in Matplotlib?
- How to manually add a legend with a color box on a Matplotlib figure ?
- How to Place Legend Outside of the Plot in Matplotlib?
- How to Remove the Legend in Matplotlib?
- Remove the legend border in Matplotlib

Line Chart

- Line chart in Matplotlib - Python

- Line plot styles in Matplotlib
- Plot Multiple lines in Matplotlib
- Change the line opacity in Matplotlib
- Increase the thickness of a line with Matplotlib
- How to Fill Between Multiple Lines in Matplotlib?

Bar Plot

- Bar Plot in Matplotlib
- Draw a horizontal bar chart with Matplotlib
- Create a stacked bar plot in Matplotlib
- Stacked Percentage Bar Plot In Matplotlib
- Plotting back-to-back bar charts Matplotlib
- How to display the value of each bar in a bar chart using Matplotlib?
- How To Annotate Bars in Barplot with Matplotlib in Python?
- How to Annotate Bars in Grouped Barplot in Python?

Histogram

- Plotting Histogram in Python using Matplotlib
- Create a cumulative histogram in Matplotlib
- How to plot two histograms together in Matplotlib?
- Overlapping Histograms with Matplotlib in Python
- Bin Size in Matplotlib Histogram

Scatter Plot

- Matplotlib Scatter
- How to add a legend to a scatter plot in Matplotlib ?
- How to Connect Scatterplot Points With Line in Matplotlib?
- How to create a Scatter Plot with several colors in Matplotlib?

- How to increase the size of scatter points in Matplotlib ?

Pie Chart

- Plot a Pie Chart in Python using Matplotlib
- How to set border for wedges in Matplotlib pie chart?
- Radially displace pie chart wedge in Matplotlib

3D Plots

- Three-dimensional Plotting in Python using Matplotlib
- 3D Scatter Plotting in Python using Matplotlib
- 3D Surface plotting in Python using Matplotlib
- 3D Wireframe plotting in Python using Matplotlib
- 3D Contour Plotting in Python using Matplotlib
- Tri-Surface Plot in Python using Matplotlib
- Surface plots and Contour plots in Python
- How to change angle of 3D plot in Python?
- How to animate 3D Graph using Matplotlib?

Working with Images

- Working with Images in Python using Matplotlib
- Python | Working with PNG Images using Matplotlib
- How to Display an Image in Grayscale in Matplotlib?
- Plot a Point or a Line on an Image with Matplotlib
- How to Draw Rectangle on Image in Matplotlib?
- How to Display an OpenCV image in Python with Matplotlib?
- Calculate the area of an image using Matplotlib

Customizing Plots

- Style Plots using Matplotlib

- Change plot size in Matplotlib - Python
- How to Change the Transparency of a Graph Plot in Matplotlib with Python?
- How to Change Fonts in matplotlib?
- How to Set Tick Labels Font Size in Matplotlib?
- How to Set Plot Background Color in Matplotlib?
- How To Adjust Position of Axis Labels in Matplotlib?
- Hide Axis, Borders and White Spaces in Matplotlib

Applications, and Projects

- 3D Sine Wave Using Matplotlib - Python
- Visualizing Bubble sort using Python
- Visualization of Merge sort using Matplotlib
- Visualization of Quick sort using Matplotlib
- 3D Visualisation of Quick Sort using Matplotlib in Python
- 3D Visualisation of Merge Sort using Matplotlib
- 3D Visualisation of Insertion Sort using Matplotlib in Python
- How to Make a Time Series Plot with Rolling Average in Python?
- Plotting Various Sounds on Graphs using Python and Matplotlib
- COVID-19 Data Visualization using matplotlib in Python
- Analyzing selling price of used cars using Python

Matplotlib Tutorial

-
-
-

Matplotlib Tutorial

Visualizing data helps us understand and share information more effectively. In PythonMatplotlibis

one of the best tools for creating visualizations. It's powerful, flexible, and lets you make many types of plots, from simple line charts to advanced animations. This tutorial will guide you step by step through using Matplotlib.

Setting Up Matplotlib

Before using Matplotlib, ensure you have it installed. You can install it using pip:

Setting Up Matplotlib

```
pip install matplotlib
```

Setting Up Matplotlib

Once installed, you can import it into your Python script:

Setting Up Matplotlib

```
import matplotlib.pyplot as plt
```

Essential Role of Pyplot in Matplotlib

Pyplot is a submodule of the Matplotlib library in Python, providing a beginner-friendly tool for creating visualizations with minimal code. It helps transform dull data into engaging and interactive plots, making it easier to analyze and draw meaningful insights for informed decision-making.

Essential Role of Pyplot in Matplotlib

Syntax:

Essential Role of Pyplot in Matplotlib

```
matplotlib.pyplot.plot()
```

Essential Role of Pyplot in Matplotlib

For more detailed information about Matplotlib Pyplot, please refer to: [Matplotlib Pyplot](#)

Creating Different Types of Plots

In data visualization, creating various types of plots is essential for effectively conveying insights from data. Below, we'll explore how to create different types of plots using Matplotlib. Each type of plot in Matplotlib is designed to address specific analytical scenarios, making them valuable tools for businesses looking for getting insights for better decision making in the industry.

1. Line Graph

Line graphs are commonly used to visualize trends over time or relationships between variables. We'll learn how to create visually appealing line graphs to represent such data.

1. Line Graph

Example

1. Line Graph

Output

1. Line Graph

Simply using `plt.plot` helped us to create an chart.

2. Bar chart

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. It can be created using the `bar()` method.

2. Bar chart

Example:

2. Bar chart

Output

3. Plotting Histogram

Histograms are basically used to represent data in the form of some groups. It is a type of bar plot where the X-axis represents the bin ranges while the Y-axis gives information about frequency. To create a histogram the first step is to create a bin of the ranges, then distribute the whole range of the values into a series of intervals, and count the values which fall into each of the intervals. Bins are clearly identified as consecutive, non-overlapping intervals of variables.

3. Plotting Histogram

Example

3. Plotting Histogram

Output

4. Scatter Plot

Scatter plots are ideal for visualizing the relationship between two continuous variables. We'll see how scatter plots help identify patterns, correlations, or clusters within data points.

4. Scatter Plot

Example:

4. Scatter Plot

4. Scatter Plot

Output

5. Pie Chart

Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges. The area of the wedge is determined by the length of the arc of the wedge.

5. Pie Chart

5. Pie Chart

Output

5. Pie Chart

You can explore many more plots according to the usecase

7. Stack Plot

Stackplots are used to create stacked area plots. It is a great way to visualize the contributions of multiple categories over time or across other continuous variables. In a stackplot, the areas representing different groups are stacked on top of each other, which allows you to see both individual and cumulative data.

7. Stack Plot

7. Stack Plot

Output

7. Stack Plot

8. Stem Plot

A stem plot, also known as a stem-and-leaf plot, is a type of plot used to display data along a number line. Stem plots are particularly useful for visualizing discrete data sets, where the values are represented as 'stems' extending from a baseline, with data points indicated as 'leaves' along the stems. Let's understand the components of a typical stem plot:

8. Stem Plot

- Stems: The stems represent the main values of the data and are typically drawn vertically along the y-axis.
- Leaves: The leaves correspond to the individual data points and are plotted horizontally along the stems.

- Baseline: The baseline serves as the reference line along which the stems are drawn.

8. Stem Plot

Example

8. Stem Plot

Output

9. Box Plot

A box plot, also known as a box-and-whisker plot, provides a visual summary of the distribution of a dataset. It represents key statistical measures such as the median, quartiles, and potential outliers in a concise and intuitive manner. Box plots are particularly useful for comparing distributions across different groups or identifying anomalies in the data.

9. Box Plot

Output

9. Box Plot

10. Error Plot

Error plots display the variability or uncertainty associated with each data point in a dataset. They are commonly used in scientific research, engineering, and statistical analysis to visualize measurement errors, confidence intervals, standard deviations, or other statistical properties of the data. By incorporating error bars into plots, we can convey not only the central tendency of the data but also the range of possible values around each point.

10. Error Plot

Output

10. Error Plot

11. Violin Plot

A violin plot is a method of visualizing the distribution of numerical data and its probability density. It is similar to a box plot but provides additional insights into the data's distribution by incorporating a kernel density estimation (KDE) plot mirrored on each side. This allows for a more comprehensive understanding of the data's central tendency, spread, and shape.

11. Violin Plot

Output

11. Violin Plot

3D Plots in Matplotlib

Sometimes, data visualization requires a three-dimensional perspective. We'll delve into creating 3D plots to visualize complex relationships and structures within multidimensional datasets.

3D Plots in Matplotlib

Output:

3D Plots in Matplotlib

The above code lets the creation of a 3D plot in Matplotlib. We can create different types of 3D plots like scatter plots, contour plots, surface plots, etc. Let's create a simple 3D line plot.

3D Plots in Matplotlib

Output

Matplotlib Subplot

Subplots are smaller axes (plots) arranged within a single figure allowing users to compare data side by side or stack visualizations for better insights. `plt.subplots()` method provides an easy way to define the layout of subplots.

Matplotlib Subplot

Syntax: `matplotlib.pyplot.subplots(nrows=1, ncols=1)`

Matplotlib Subplot

This syntax creates a figure with `nrows` rows and `ncols` columns of subplots.

Matplotlib Subplot

For more detailed information about Matplotlib Subplot , Refer to the Article [Matplotlib subplot](#)

Matplotlib Subplot

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Matplotlib Subplot

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

Matplotlib Subplot

- AI-ML-DS
- Data Visualization
- Python-matplotlib

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media

- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS

- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git

- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter

- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course

- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification

- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Creating a plot

- Line Chart using Plotly in Python
- Bar chart using Plotly in Python
- Histogram using Plotly in Python
- Scatter plot using Plotly in Python
- Bubble chart using Plotly in Python
- Pie plot using Plotly in Python
- Box Plot using Plotly in Python
- Gantt Chart in plotly
- Contour Plots using Plotly in Python
- Create Heatmaps using graph_objects class in Plotly
- Sunburst Plot using Plotly in Python
- Polar Charts using Plotly in Python
- Ternary Plots in Plotly
- 3D Line Plots using Plotly in Python
- 3D Surface Plots using Plotly in Python
- 3D Bubble chart using Plotly in Python

- 3D Mesh Plots using Plotly in Python

Advanced Plot Customization

- How to display image using Plotly?
- Python Plotly - Subplots and Inset Plots
- Hover Text and Formatting in Python-Plotly
- How to apply different titles for each different subplots using Plotly in Python?

Plotly Dash Basics

- Introduction to Dash in Python
- Plot Live Graphs using Python Dash and Plotly
- How to Install Python Dash with Conda ?

Plotly tutorial

-
-
-

Plotly tutorial

Plotlylibrary in Python is an open-source library that can be used for data visualization and understanding data simply and easily. Plotly supports various types of plots like line charts, scatter plots, histograms, box plots, etc. So you all must be wondering why Plotly is over other visualization tools or libraries. So here are some reasons :

Plotly tutorial

- Plotly has hover tool capabilities that allow us to detect any outliers or anomalies in a large number of data points.
- It is visually attractive and can be accepted by a wide range of audiences.
- Plotly generally allows us endless customization of our graphs and makes our plot more

meaningful and understandable for others.

Plotly tutorial

Plotly tutorial

This tutorial aims at providing you the insight about Plotly with the help of the huge dataset explaining the Plotly from basics to advance and covering all the popularly used charts.

Plotly tutorial

Table of Content

Plotly tutorial

- How to install Plotly?
- Package Structure of Plotly
- Getting Started
- Creating Different Types of Charts
- Heatmaps
- Error Bars
- 3D Line Plots
- 3D Scatter Plot Plotly
- 3D Surface Plots
- Interacting with the Plots
- Adding Buttons to the Plot
- Creating Sliders and Selectors to the Plot
- More Plots using Plotly

How to install Plotly?

Before installing Plotly in system, you need to install pip in your system, Refer to ?

How to install Plotly?

Download and install pip Latest Version

How to install Plotly?

Plotly does not come built-in with Python. To install it type the below command in the terminal.

How to install Plotly?

How to install Plotly?

This may take some time as it will install the dependencies as well.

Package Structure of Plotly

There are three main modules in Plotly. They are:

Package Structure of Plotly

- plotly.plotly
- plotly.graph_objects
- plotly.tools

Package Structure of Plotly

plotly.plotly acts as the interface between the local machine and Plotly. It contains functions that require a response from Plotly's server.

Package Structure of Plotly

plotly.graph_objects module contains the objects (Figure, layout, data, and the definition of the plots like scatter plot, line chart) that are responsible for creating the plots. The Figure can be represented either as dict or instances of plotly.graph_objects.Figure and these are serialized as JSON before it gets passed to plotly.js. Consider the below example for better understanding.

Package Structure of Plotly

Note: plotly.express module can create the entire Figure at once. It uses the graph_objects internally and returns the graph_objects.Figure instance.

Package Structure of Plotly

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px # Creating the Figure instance
fig = px.line(x=[1,2,3], y=[1,2,3]) # printing the figure instance
print(fig)
```

Python3

Output:

Python3

Figures are represented as trees where the root node has three top layer attributes ?data, layout, and frames and the named nodes called ?attributes?. Consider the above example, layout.legend is a nested dictionary where the legend is the key inside the dictionary whose value is also a dictionary.

Python3

plotly.tools module contains various tools in the forms of the functions that can enhance the Plotly experience.

Getting Started

After learning the installation and basic structure of the Plotly, let's create a simple plot using the pre-defined data sets defined by the plotly.

Getting Started

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px # Creating the Figure instance
fig = px.line(x=[1,2,3], y=[1,2,3]) # showing the plot
fig.show()
```

Python3

Output:

Python3

Python3

In the above example, the `plotly.express` module is imported which returns the Figure instance. We have created a simple line chart by passing the x, y coordinates of the points to be plotted.

Creating Different Types of Charts

With `plotly` we can create more than 40 charts and every plot can be created using the `plotly.express` and `plotly.graph_objects` class. Let's see some commonly used charts with the help of `Plotly`.

Line Chart

Line plot in `Plotly` is much accessible and illustrious annexation to `plotly` which manage a variety of types of data and assemble easy-to-style statistic. With `px.line` each data position is represented as a vertex (which location is given by the x and y columns) of a polyline mark in 2D space.

Line Chart

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the iris datasetdf=px.data.iris()# plotting the line chartfig=px.line(df, x="species", y="petal_width")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the line charts.

Python3

- plotly.express.line() function in Python
- Line Chart using Plotly in Python

Bar Chart

A bar chart is a pictorial representation of data that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. In other words, it is the pictorial representation of a dataset. These data sets contain the numerical values of variables that represent the length or height.

Bar Chart

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px # using the iris dataset
df=px.data.iris() # plotting the bar chart
fig=px.bar(df, x="sepal_width", y="sepal_length") # showing the plot
fig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the bar chart.

Python3

- Bar chart using Plotly in Python
- How to create Stacked bar chart in Python-Plotly?
- How to group Bar Charts in Python-Plotly?

Histograms

A histogram contains a rectangular area to display the statistical information which is proportional to the frequency of a variable and its width in successive numerical intervals. A graphical representation that manages a group of data points into different specified ranges. It has a special feature that shows no gaps between the bars and similar to a vertical bar graph.

Histograms

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the iris datasetdf=px.data.iris()# plotting the histogramfig=px.histogram(df, x="sepal_length", y="petal_width")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the histograms.

Python3

- Histogram using Plotly in Python
- Histograms in Plotly using graph_objects class
- How to create a Cumulative Histogram in Plotly?

Scatter Plot and Bubble charts

A scatter plot is a set of dotted points to represent individual pieces of data in the horizontal and vertical axis. A graph in which the values of two variables are plotted along X-axis and Y-axis, the pattern of the resulting points reveals a correlation between them.

Scatter Plot and Bubble charts

A bubble plot is a scatter plot with bubbles (color-filled circles). Bubbles have various sizes dependent on another variable in the data. It can be created using the scatter() method of plotly.express.

Scatter Plot and Bubble charts

Example 1:Scatter Plot

Python3

Python3

Python3

Python3

```
importplotly.express as px# using the iris datasetdf=px.data.iris()# plotting the scatter chartfig=px.scatter(df, x="species", y="petal_width")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Example 2:Bubble Plot

Python3

Python3

Python3

Python3

```
importplotly.express as px# using the iris datasetdf=px.data.iris()# plotting the bubble chartfig=px.scatter(df, x="species", y="petal_width",size="petal_length", color="species")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the scatter plots and bubble plots.

Python3

- `plotly.express.scatter()` function in Python
- Scatter plot in Plotly using `graph_objects` class
- Scatter plot using Plotly in Python
- Bubble chart using Plotly in Python

Pie Charts

A pie chart is a circular statistical graphic, which is divided into slices to illustrate numerical proportions. It depicts a special chart that uses 'pie slices', where each sector shows the relative sizes of data. A circular chart cuts in a form of radii into segments describing relative frequencies or magnitude also known as circle graph.

Pie Charts

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the tips datasetdf=px.data.tips()# plotting the pie chartfig=px.pie(df, values="total_bill", names="day")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the pie charts.

Python3

- Pie plot using Plotly in Python

Box Plots

A Box Plot is also known as Whisker plot. It is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

Box Plots

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the tips datasetdf=px.data.tips()# plotting the box chartfig=px.box(df, x="day", y="total_bill")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about box plots.

Python3

- Box Plot using Plotly in Python
- Box plot in Plotly using graph_objects class
- How to create Grouped box plot in Plotly?

Violin plots

A Violin Plot is a method to visualize the distribution of numerical data of different variables. It is similar to Box Plot but with a rotated plot on each side, giving more information about the density estimate on the y-axis. The density is mirrored and flipped over and the resulting shape is filled in, creating an image resembling a violin. The advantage of a violin plot is that it can show nuances in the distribution that aren't perceptible in a boxplot. On the other hand, the boxplot more clearly shows the outliers in the data.

Violin plots

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the tips datasetdf=px.data.tips()# plotting the violin chartfig=px.violin(df, x="day", y="total_bill")# showing the plotfig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the violin plots

Python3

- Violin Plots using Plotly

Gantt Charts

Generalized Activity Normalization Time Table (GANTT) chart is type of chart in which series of horizontal lines are present that show the amount of work done or production completed in given period of time in relation to amount planned for those projects.

Gantt Charts

Example:

Python3

Python3

Python3

Python3

```
import plotly.figure_factory as ff# Data to be plotteddf=[dict(Task="A", Start='2020-01-01', Finish='2020-02-02'),dict(Task="Job B", Start='2020-03-01', Finish='2020-11-11'),dict(Task="Job C", Start='2020-08-06', Finish='2020-09-21')]# Creating the plotfig=ff.create_gantt(df)fig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the Gantt Charts.

Python3

- Gantt Chart in Plotly

Contour Plots

A Contour plots also called level plots are a tool for doing multivariate analysis and visualizing 3-D plots in 2-D space. If we consider X and Y as our variables we want to plot then the response Z will be plotted as slices on the X-Y plane due to which contours are sometimes referred as Z-slices or iso-response.

Contour Plots

A contour plots is used in the case where you want to see the changes in some value (Z) as a function with respect to the two values (X, Y). Consider the below example.

Contour Plots

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as go # Creating the X, Y value that will# change the values of Z as a
function
feature_x=np.arange(0,50,2)
feature_y=np.arange(0,50,3) # Creating 2-D grid of features
[X,
Y]=np.meshgrid(feature_x, feature_y)
Z=np.cos(X/2)+np.sin(Y/4) # plotting the
```

```
figurefig=go.Figure(data=go.Contour(x=feature_x, y=feature_y, z=Z))fig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about contour plots.

Python3

- Contour Plots using Plotly in Python

Heatmaps

Heatmaps are defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred.

Heatmap is also defined by the name of the shading matrix.

Heatmaps

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as go
feature_x=np.arange(0,50,2)
feature_y=np.arange(0,50,3)
# Creating 2-D grid of features
[X, Y]=np.meshgrid(feature_x, feature_y)
Z=np.cos(X/2)+np.sin(Y/4)
# plotting the figure
fig=go.Figure(data=go.Heatmap(x=feature_x, y=feature_y, z=Z,))fig.show()
```


Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the heatmaps.

Python3

- Create Heatmaps using graph_objects class in Plotly
- Annotated Heatmaps using Plotly in Python

Error Bars

For functions representing 2D data points such as `px.scatter`, `px.line`, `px.bar`, etc., error bars are given as a column name which is the value of the `error_x` (for the error on x position) and `error_y` (for the error on y position). Error bars are the graphical presentation alternation of data and used on graphs to imply the error or uncertainty in a reported capacity.

Error Bars

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px# using the iris datasetdf=px.data.iris()# Calculating the error fielddf["error"]=df["petal_length"]/100# plotting the scatter chartfig=px.scatter(df, x="species", y="petal_width",error_x="error", error_y="error")# showing the plotfig.show()
```

Python3

Output:

Python3

3D Line Plots

Line plot in plotly is much accessible and illustrious annexation to plotly which manage a variety of types of data and assemble easy-to-style statistic. With `px.line_3d` each data position is represented as a vertex (which location is given by the x, y and z columns) of a polyline mark in 3D space.

3D Line Plots

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px # data to be plotted
df = px.data.tips() # plotting the figure
fig = px.line_3d(df,
x="sex", y="day", z="time", color="sex")
fig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the 3D line charts.

Python3

- plotly.express.line_3d() function in Python

- 3D Line Plots using Plotly in Python

3D Scatter Plot Plotly

3D Scatter Plot can plot two-dimensional graphics that can be enhanced by mapping up to three additional variables while using the semantics of hue, size, and style parameters. All the parameter control visual semantic which are used to identify the different subsets. Using redundant semantics can be helpful for making graphics more accessible. It can be created using the scatter_3d function of plotly.express class.

3D Scatter Plot Plotly

Example:

Python3

Python3

Python3

Python3

```
import plotly.express as px # Data to be plotted
df = px.data.iris() # Plotting the figure
fig = px.scatter_3d(df, x='sepal_width', y='sepal_length', z='petal_width', color='species')
fig.show()
```

Python3

Output:

Python3

Python3

Refer to the below articles to get detailed information about the 3D scatter plot.

Python3

- 3D scatter plot using Plotly in Python
- 3D Scatter Plot using graph_objects Class in Plotly-Python
- 3D Bubble chart using Plotly in Python

3D Surface Plots

Surface plots are those plots which have three-dimensional data, which are X, Y, and Z. Rather than showing individual data points, the surface plot has a functional relationship between dependent variable Y and has two independent variables X and Z. This plot is used to distinguish between dependent and independent variables.

3D Surface Plots

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as go
import numpy as np
# Data to be plotted
x=np.linspace(-2,2,30), np.ones(30))
y=x.copy()
z=np.cos(x**2+y**2)
# plotting the figure
fig=go.Figure(data=[go.Surface(x=x, y=y, z=z)])
fig.show()
```

Python3

Output:

Python3

Interacting with the Plots

Plotly provides various tools for interacting with the plots such as adding dropdowns, buttons,

sliders, etc. These can be created using the update menu attribute of the plot layout. Let's see how to do all such things in detail.

Creating Dropdown Menu in Plotly

Adrop-down menus a part of the menu-button which is displayed on a screen all the time. Every menu button is associated with a Menu widget that can display the choices for that menu button when clicked on it. In plotly, there are 4 possible methods to modify the charts by using update menu method.

Creating Dropdown Menu in Plotly

- restyle:modify data or data attributes
- layout:modify layout attributes
- update:modify data and layout attributes
- animate:start or pause an animation

Creating Dropdown Menu in Plotly

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as px
import numpy as np # creating random data through randint#
function of numpy.random.seed(42)# Data to be
Plotted
random_x=np.random.randint(1,101,100)
random_y=np.random.randint(1,101,100)
plot=px.Figure(data=[px.Scatter(x=random_x,y=random_y,mode='markers',)])# Add
dropdown
plot.update_layout(update_menus=[dict(buttons=list([dict(args=["type","scatter"],label="Scat
```

```
ter                                Plot",method="restyle"),dict(args=["type","bar"],label="Bar  
Chart",method="restyle"))],direction="down",),,])plot.show()
```

Python3

Output:

Python3

Python3

Output

Python3

Python3

In the above example we have created two graphs for the same data. These plots are accessible using the dropdown menu.

Adding Buttons to the Plot

In plotly,actions custom Buttonsare used to quickly make actions directly from a record. Custom Buttons can be added to page layouts in CRM, Marketing, and Custom Apps. There are also 4 possible methods that can be applied in custom buttons:

Adding Buttons to the Plot

- restyle:modify data or data attributes
- relayout:modify layout attributes
- update:modify data and layout attributes
- animate:start or pause an animation

Adding Buttons to the Plot

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as px
import pandas as pd

pddf = go.data.tips()
plot = px.Figure(data=[px.Scatter(x=data['day'], y=data['tip'], mode='markers')])

# Add dropdown
plot.update_layout(
    updatemenus=[dict(
        type="buttons",
        direction="left",
        buttons=list([
            dict(
                args=["type", "scatter"],
                label="Scatter Plot",
                method="restyle"
            ),
            dict(
                args=["type", "bar"],
                label="Bar Chart",
                method="restyle"
            )
        ])
    )]
)
plot.show()
```

Python3

Output:

Python3

Python3

In this example also we are creating two different plots on the same data and both plots are accessible by the buttons.

Creating Sliders and Selectors to the Plot

In plotly, the range slider is a custom range-type input control. It allows selecting a value or a range of values between a specified minimum and maximum range. And the range selector is a tool for selecting ranges to display within the chart. It provides buttons to select pre-configured ranges in the chart. It also provides input boxes where the minimum and maximum dates can be manually input.

Creating Sliders and Selectors to the Plot

Example:

Python3

Python3

Python3

Python3

```
import plotly.graph_objects as px
import plotly.express as go
import numpy as np
pdf = go.data.tips()
x = df['total_bill']
y = df['day']
plot = px.Figure(data=[px.Scatter(x=x, y=y, mode='lines',)])
plot.update_layout(xaxis=dict(rangeslider=dict(buttons=list([dict(count=1, step="day", stepmode="backward"),])), rangeslider=dict(visible=True),))
plot.show()
```

Python3

Output:

Python3

Python3

Output

Python3

More Plots using Plotly

- plotly.express.scatter_geo() function in Python
- plotly.express.scatter_polar() function in Python
- plotly.express.scatter_ternary() function in Python
- plotly.express.line_ternary() function in Python
- Filled area chart using plotly in Python
- How to Create Stacked area plot using Plotly in Python?
- Sunburst Plot using Plotly in Python
- Sunburst Plot using graph_objects class in plotly

- `plotly.figure_factory.create_annotated_heatmap()` function in Python
- `plotly.figure_factory.create_2d_density()` function in Python
- Ternary contours Plot using Plotly in Python
- How to make Log Plots in Plotly ? Python?
- Polar Charts using Plotly in Python
- Carpet Contour Plot using Plotly in Python
- Ternary Plots in Plotly
- How to create a Ternary Overlay using Plotly?
- Parallel Coordinates Plot using Plotly in Python
- Carpet Plots using Plotly in Python
- 3D Cone Plots using Plotly in Python
- 3D Volume Plots using Plotly in Python
- 3D Streamtube Plots using Plotly in Python
- 3D Mesh Plots using Plotly in Python
- How to create Tables using Plotly in Python?
- `plotly.figure_factory.create_dendrogram()` function in Python
- Define Node position in Sankey Diagram in plotly
- Sankey Diagram using Plotly in Python
- Quiver Plots using Plotly in Python
- Treemap using Plotly in Python
- Treemap using `graph_objects` class in plotly
- `plotly.figure_factory.create_candlestick()` function in Python
- `plotly.figure_factory.create_choropleth()` function in Python
- `plotly.figure_factory.create_bullet()` in Python
- Streamline Plots in Plotly using Python
- How to make Wind Rose and Polar Bar Charts in Plotly ? Python?
- Title alignment in Plotly

- Change marker border color in Plotly ? Python
- Plot Live Graphs using Python Dash and Plotly
- Animated Data Visualization using Plotly Express
- Introduction to Plotly-online using Python
- How to display image using Plotly?

More Plots using Plotly

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the Three 90 Challenge?complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

More Plots using Plotly

- Python
- Python-Plotly

More Plots using Plotly

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution

- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures

- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs

- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker

- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy

- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced

- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course

- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE

- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is python scikit library?

-
-
-

What is python scikit library?

Python is known for its versatility across various domains, from web development to data science and machine learning. In machine learning, one of the go-to libraries for Python enthusiasts is Scikit-learn, often referred to as "sklearn." It's a powerhouse for creating robust machine learning models.

What is Scikit-learn Library?

Scikit-learn is an open-source machine learning library that provides simple and efficient tools for data analysis and modeling. It is built on NumPy, SciPy, and Matplotlib, making it a powerful tool for tasks like classification, regression, clustering, and dimensionality reduction.

What is Scikit-learn Library?

- Classification: Classification involves teaching a computer to categorize things. For example, a model could be built to determine whether an email is spam or not.
- Regression: Regression predicting numbers based on other numbers. For instance, a model could predict house prices using factors like location, size, and age.
- Clustering: Clustering involves finding patterns in data and grouping similar items together. For

example, customers could be segmented into different groups based on their shopping habits.

- Dimensionality Reduction: Dimensionality reduction helps focus on essential data parts while discarding noise. This is useful when dealing with a lot of data that isn't all relevant.

Features of Scikit-Learn

Scikit-learn is indeed a versatile tool for machine learning tasks, offering a wide range of features to address various aspects of the data science pipeline. let's examine prime key features of scikit-learn:

Supervised Learning

- Classification: Algorithms for predicting categorical labels, including logistic regression, decision trees, random forests, support vector machines (SVMs) and gradient boosting.
- Regression: Algorithms for predicting continuous outputs, including linear regression, support vector regression, and decision tree regression.

Unsupervised Learning

- Clustering: Techniques for grouping data points into similar clusters, including K-means clustering, DBSCAN, and hierarchical clustering.
- Dimensionality Reduction: Methods for reducing the number of features in your data, such as principal component analysis (PCA).

Data Preprocessing

- Data Splitting: Functions to split your data into training and testing sets for model evaluation.
- Feature Scaling: Techniques for normalizing the scale of your features.
- Feature Selection: Methods to identify and select the most relevant features for your model.
- Feature Extraction: Tools to create new features from existing ones, such as text vectorization for natural language processing tasks.

Model Evaluation

- Metrics: Functions to calculate performance metrics like accuracy, precision, recall, and F1-score for classification models, and mean squared error (MSE) for regression models.
- Model Selection: Tools for selecting the best model hyperparameters through techniques like grid search and randomized search.

Additional Features

- Inbuilt datasets: Scikit-learn provides a variety of sample datasets for experimentation and learning purposes.
- Easy to Use API: Scikit-learn is known for its consistent and user-friendly API, making it accessible to both beginners and experienced data scientists.
- Open Source: Scikit-learn is an open-source library with a large and active community, ensuring continuous development and support.

Implementation of Scikit Library in Python

Steps for implementing Scikit-learn in Python:

Implementation of Scikit Library in Python

- Installation: First, you need to install Scikit-learn if you haven't already. You can install it using pip, Python's package manager, with the following command:

Implementation of Scikit Library in Python

- Importing: Once installed, you can import Scikit-learn modules into your Python script or environment using the import statement. For example:

Classification - Logistic Regression Algorithm Example

Logistic Regression is a binary classification algorithm that estimates probabilities of a binary outcome. It's used for problems like spam detection, medical diagnosis, and credit scoring. It's chosen for its simplicity, interpretability, and effectiveness in linearly separable datasets.

Classification - KNN Classifier Algorithm Example

K-Nearest Neighbors (KNN) algorithm classifies data points based on the majority class of their nearest neighbors. It's useful for simple classification tasks, particularly when data is not linearly separable or when decision boundaries are complex. It's used in recommendation systems, handwriting recognition, and medical diagnosis.

Regression - Linear Regression Algorithm Example

Linear Regression fits a linear model to observed data points, predicting continuous outcomes based on input features. It's used when exploring relationships between variables and making predictions. Applications include economics, finance, engineering, and social sciences.

Clustering - KMeans Algorithm Example

KMeans algorithm partitions data into k clusters based on similarity. It's used for unsupervised clustering tasks like customer segmentation, image compression, and anomaly detection. Ideal when data's structure is unknown but grouping is desired.

Dimensionality Reduction - PCA Example

PCA (Principal Component Analysis) reduces the dimensionality of data by finding the most important features. It's used for visualizing high-dimensional data, noise reduction, and speeding up machine learning algorithms. Commonly applied in image processing, genetics, and finance.

Advantages of scikit library

- Easy to Use: Simple and user-friendly interface for machine learning tasks.
- Extensive Algorithm Support: Offers a wide range of algorithms for various tasks like classification, regression, clustering, and more.
- Data Preprocessing Tools: Provides tools for data preprocessing, including scaling, normalization, and handling missing values.
- Model Evaluation: Offers metrics for evaluating model performance and techniques like cross-validation for robust assessment.
- Integration: Integrates well with other Python libraries like NumPy, Pandas, and Matplotlib.

Disadvantages of scikit library

- Limited Deep Learning Support: Doesn't have extensive support for deep learning algorithms compared to specialized libraries like TensorFlow or PyTorch.
- Scaling with Large Datasets: May face performance issues with very large datasets due to its single-machine architecture.
- Complex Model Customization: Customizing complex model architectures or implementing new algorithms may require additional coding outside Scikit-learn.

Conclusion

Scikit-learn stands out as a powerful and versatile machine learning library for Python developers. Its ease of use, extensive algorithm support, and robust tools for data preprocessing and model evaluation make it a go-to choice for both beginners and experts in the field.

Conclusion

While it has limitations such as limited deep learning support and scalability challenges with large datasets, its applications in classification, regression, clustering, dimensionality reduction, and model evaluation showcase its relevance across a wide range of machine learning tasks.

Conclusion

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Conclusion

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Conclusion

T

Conclusion

- Machine Learning
- AI-ML-DS
- AI-ML-DS With Python

Conclusion

- Machine Learning

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner

- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL

- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development

- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements

- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is PyTorch ?

-
-

-

What is PyTorch ?

Deep Learning is a branch of Machine Learning where algorithms are written that mimic the functioning of a human brain. The most commonly used libraries in deep learning are Tensorflow and PyTorch. PyTorch is an open-source deep learning framework available with a Python and C++ interface. The PyTorch resides inside the torch module. In PyTorch, the data that has to be processed is input in the form of a tensor.

How to Install Pytorch?

If you have Anaconda Python Package manager installed in your system, then using by running the following command in the terminal will install PyTorch:

How to Install Pytorch?

This command will install the latest Stable version of PyTorch. Stable represents the most currently tested and supported version of PyTorch. The latest stable version of PyTorch is 1.12.1. If you want to use PyTorch without explicitly installing it on your local machine, you can use Google Colab.

How PyTorch Works?

Here is an overview of How PyTorch Works ?

How PyTorch Works?

Tensors: At the core of PyTorch is the torch.Tensor class, which is a multi-dimensional array similar to NumPy arrays.

Python3

Python3

Python3

Python3

```
import torch # Create a tensor x = torch.tensor([1, 2, 3])
```

Python3

Python3

Dynamic Computational Graphs: In PyTorch, the graph is built on-the-fly as operations are performed, allowing for dynamic changes during runtime. This is in contrast to static computation graphs used by some other deep learning frameworks.

Python3

Python3

Python3

Python3

```
# Dynamic computation y = x + 2z = y * 3
```

Python3

Python3

Autograd: It automatically computes gradients of tensors with respect to some scalar value. This is crucial for training neural networks using gradient-based optimization algorithms.

Python3

Python3

Python3

Python3

```
z.backward() # Compute gradients print(x.grad) # Access gradients
```

Python3

Python3

Neural Network Module: PyTorch provides the `torch.nn` module, which includes pre-defined layers, loss functions, and other components for building neural networks. Users can also define custom neural network architectures by subclassing the `torch.nn.Module` class.

Python3

Python3

Python3

Python3

```
import torch.nn as nn # Define a simple neural network
class SimpleNN(nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        self.fc = nn.Linear(10, 5)
    def forward(self, x):
        return self.fc(x)
```

PyTorch tensors

The PyTorch is used to process the tensors. Tensors are multidimensional arrays like n-dimensional NumPy array. However, tensors can be used in GPUs as well, which is not in the case of NumPy array. PyTorch accelerates the scientific computation of tensors as it has various inbuilt functions.

PyTorch tensors

A vector is a one-dimensional tensor, and a matrix is a two-dimensional tensor. One significant difference between the Tensor and multidimensional array used in C, C++, and Java is tensors should have the same size of columns in all dimensions. Also, the tensors can contain only numeric data types.

PyTorch tensors

PyTorch tensors

The two fundamental attributes of a tensor are:

PyTorch tensors

- Shape: refers to the dimensionality of array or matrix
- Rank: refers to the number of dimensions present in tensor

PyTorch tensors

Example 1:

Python3

Python3

Python3

Python3

```
# importing torchimport torch# creating a tensorst1=torch.tensor([1,2,3,4])t2=torch.tensor([[1,2,3,4],[5,6,7,8],[9,10,11,12]])# printing the tensors:print("Tensor t1: \n", t1)print("\nTensor t2: \n", t2)# rank of tensorsprint("\nRank of t1: ",len(t1.shape))print("Rank of t2: ",len(t2.shape))# shape of tensorsprint("\nRank of t1: ", t1.shape)print("Rank of t2: ", t2.shape)
```

Python3

Output:

Creating Tensor in PyTorch

There are various methods to create a tensor in PyTorch. A tensor can contain elements of a single data type. We can create a tensor using a python list or NumPy array. The torch has 10 variants of tensors for both GPU and CPU. Below are different ways of defining a tensor.

Creating Tensor in PyTorch

- `torch.Tensor()` :It copies the data and creates its tensor. It is an alias for `torch.FloatTensor`.
- `torch.tensor()` :It also copies the data to create a tensor; however, it infers the data type automatically.
- `torch.as_tensor()` :The data is shared and not copied in this case while creating the data and accepts any type of array for tensor creation.
- `torch.from_numpy()` :It is similar to `tensor.as_tensor()` however it accepts only numpy array.

Creating Tensor in PyTorch

Example 2:

Python3

Python3

Python3

Python3

```
# importing torch moduleimporttorchimportnumpy as np# list of values to be stored as tensorsdata1=[1,2,3,4,5,6]data2=np.array([1.5,3.4,6.8,9.3,7.0,2.8])# creating tensors and printingt1=torch.tensor(data1)t2=torch.Tensor(data1)t3=torch.as_tensor(data2)t4=torch.from_numpy(data2)print("Tensor: ",t1,"Data type: ", t1.dtype,"\n")print("Tensor: ",t2,"Data type: ", t2.dtype,"\n")print("Tensor: ",t3,"Data type: ", t3.dtype,"\n")print("Tensor: ",t4,"Data type: ", t4.dtype,"\n")
```

Python3

Output:

Restructuring Tensors in Pytorch

We can modify the shape and size of a tensor as desired in PyTorch. We can also create a

transpose of an n-d tensor. Below are three common ways to change the structure of your tensor as desired:

Restructuring Tensors in Pytorch

`.reshape(a, b)` :returns a new tensor with size a,b

Restructuring Tensors in Pytorch

`.resize(a, b)` :returns the same tensor with the size a,b

Restructuring Tensors in Pytorch

`.transpose(a, b)` :returns a tensor transposed in a and b dimension

Restructuring Tensors in Pytorch

Restructuring Tensors in Pytorch

A 2*3 matrix has been reshaped and transposed to 3*2. We can visualize the change in the arrangement of the elements in the tensor in both cases.

Restructuring Tensors in Pytorch

Example 3:

Python3

Python3

Python3

Python3

```
# import torch moduleimporttorch# defining tensor=torch.tensor([[1,2,3,4],[5,6,7,8],[9,10,11,12]])#  
reshaping      the      tensorprint("Reshaping")print(t.reshape(6,2))#      resizing      the  
tensorprint("\nResizing")print(t.resize(2,6))#      transposing      the  
tensorprint("\nTransposing")print(t.transpose(1,0))
```

Python3

Output:

Mathematical Operations on Tensors in PyTorch

We can perform various mathematical operations on tensors using Pytorch. The code for performing Mathematical operations is the same as in the case with NumPy arrays. Below is the code for performing the four basic operations in tensors.

Mathematical Operations on Tensors in PyTorch

Example 4:

Python3

Python3

Python3

Python3

```
# import torch moduleimport torch# defining two tensors
t1=torch.tensor([1,2,3,4])t2=torch.tensor([5,6,7,8])# adding two tensors
print("tensor2 + tensor1")print(torch.add(t2, t1))# subtracting two tensor
print("\ntensor2 - tensor1")print(torch.sub(t2, t1))# multiplying two tensors
print("\ntensor2 * tensor1")print(torch.mul(t2, t1))# diving two tensors
print("\ntensor2 / tensor1")print(torch.div(t2, t1))
```

Python3

Output:

Python3

For getting into further in-depth matrix multiplication using Pytorch . You can refer to this article ?

Python3

Python ? Matrix multiplication using Pytorch

Pytorch Modules

The PyTorch library modules are essential to create and train neural networks. The three main library modules are Autograd, Optim, and nn.

Pytorch Modules

1. Autograd Module: The autograd provides the functionality of easy calculation of gradients without the explicitly manual implementation of forward and backward pass for all layers.

Pytorch Modules

For training any neural network we perform backpropagation to calculate the gradient. By calling the `.backward()` function we can calculate every gradient from the root to the leaf.

Pytorch Modules

Example 5:

Python3

Python3

Python3

Python3

```
# importing torchimport torch# creating a tensort1=torch.tensor(1.0,requires_grad=True)t2=torch.tensor(2.0, requires_grad=True)# creating a variable and gradientz=100*t1*t2z.backward()# printing gradientprint("dz/dt1 : ", t1.grad.data)print("dz/dt2 : ", t2.grad.data)
```

Python3

Output:

Python3

2. Optim Module:PyTorch Optium Module which helps in the implementation of various optimization algorithms. This package contains the most commonly used algorithms like Adam, SGD, and RMS-Prop. To use torch.optim we first need to construct an Optimizer object which will keep the parameters and update it accordingly. First, we define the Optimizer by providing the optimizer algorithm we want to use. We set the gradients to zero before backpropagation. Then for updation of parameters the optimizer.step() is called.

Python3

```
optimizer = torch.optim.Adam(model.parameters(), lr=0.01) #defining optimizer
```

Python3

```
optimizer.zero_grad() #setting gradients to zero
```

Python3

```
optimizer.step() #parameter updation
```

Python3

3. nn Module:This package helps in the construction of neural networks. It is used to build layers.

Python3

For creating a model with a single layer we can simply define it by using nn.Sequential().

Python3

```
model = nn.Sequential( nn.Linear(in, out), nn.Sigmoid(), nn.Linear(_in, _out), nn.Sigmoid() )
```

Python3

For the implementation of the model which is not in a single sequence, we define a model by subclassing the nn. Module class.

Python3

Example 6:

Python3

Python3

Python3

Python3

```
class Model(nn.Module):  
    def __init__(self):  
        super(Model, self).__init__()  
        self.linear = torch.nn.Linear(1, 1)  
    def forward(self, x):  
        y_pred = self.linear(x)  
        return y_pred
```

PyTorch Dataset and Dataloader

The `torch.utils.data.Dataset` class contains all the custom datasets. We need to implement two methods, `__len__()` and `__getitem__()`, to create our own dataset class. The PyTorch Dataloader has an amazing feature of loading the dataset in parallel with automatic batching. It, therefore, reduces the time of loading the dataset sequentially hence enhancing the speed.

PyTorch Dataset and Dataloader

Syntax: `Dataloader(dataset, shuffle=True, sampler=None, batch_sampler=None, batch_size=32)`

PyTorch Dataset and Dataloader

The PyTorch DataLoader supports two types of datasets:

PyTorch Dataset and Dataloader

- Map-style datasets: The data items are mapped to the indexes. In these datasets, the `__getitem__()` method is used to retrieve the index of each item.
- Iterable-style datasets: In these datasets `__iter__()` protocol is implemented. The data samples are retrieved in sequence.

PyTorch Dataset and Dataloader

Please refer to the article on [Using a DataLoader in PyTorch](#) to know more.

Building Neural Network with PyTorch

We will see this in a stepwise implementation:

Building Neural Network with PyTorch

- Dataset Preparation: As everything in PyTorch is represented in the form of tensors, so we should first in tensors.
- Building model: For building a neural network first we first define the number of input layers, hidden layers, and output layers. We also need to define the initial weights. The values of the weights matrices are chosen randomly using `torch.randn()`. `Torch.randn()` returns a tensor consisting of random numbers from a standard normal distribution.
- Forward Propagation: The data is fed to a neural network and a matrix multiplication is performed between the weights and input. This can be easily done using a torch.
- Loss computation: PyTorch.nn functions have multiple loss functions. Loss functions are used to measure the error between the predicted value to the target value.
- Backpropagation: It is used for optimizing weights. The weights are changed such that the loss is minimized.

Building Neural Network with PyTorch

Now let's build a neural network from scratch:

Building Neural Network with PyTorch

Example 7:

Python3

Python3

Python3

Python3

```
# importing torchimporttorch# training input(X) and output(y)X=torch.Tensor([[1], [2], [3],[4], [5], [6]])y=torch.Tensor([[5], [10], [15],[20], [25], [30]])classModel(torch.nn.Module):# defining layerdef__init__(self):super(Model,self).__init__()self.linear=torch.nn.Linear(1,1)# implementing forward passdefforward(self, x):y_pred=self.linear(x)returny_predmodel=torch.nn.Linear(1,1)# defining loss function and optimizerloss_fn=torch.nn.L1Loss()optimizer=torch.optim.Adam(model.parameters(), lr=0.01)forepochinrange(1000):# predicting y using initial weightsy_pred=model(X.requires_grad_())# loss calculationloss=loss_fn(y_pred, y)# calculating gradientsloss.backward()# updating weightsoptimizer.step()optimizer.zero_grad()# testing on new dataX=torch.Tensor([[7], [8]])predicted=model(X)print(predicted)
```

Python3

Output:

Python3

Python3

Get IBM Certificationand a90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

Python3

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need?Start the challenge right away!

Python3

- AI-ML-DS

- Deep Learning

- Python-PyTorch

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy

- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Data Analysis

- What is Data Analysis?
- Data Analytics and its type
- How to Install Numpy on Windows?
- How to Install Pandas in Python?
- How to Install Matplotlib on python?
- How to Install Python Tensorflow in Windows?

Data Analysis Libraries

- Pandas Tutorial
- NumPy Tutorial - Python Library
- Data Analysis with SciPy
- Introduction to TensorFlow

Data Visualization Libraries

- Matplotlib Tutorial
- Python Seaborn Tutorial
- Plotly tutorial
- Introduction to Bokeh in Python

Exploratory Data Analysis (EDA)

- Univariate, Bivariate and Multivariate data and its analysis
- Measures of Central Tendency in Statistics
- Measures of Spread - Range, Variance, and Standard Deviation
- Interquartile Range and Quartile Deviation using NumPy and SciPy
- Anova Formula
- Skewness of Statistical Data
- How to Calculate Skewness and Kurtosis in Python?
- Difference Between Skewness and Kurtosis
- Histogram | Meaning, Example, Types and Steps to Draw
- Interpretations of Histogram
- Box Plot
- Quantile Quantile plots
- What is Univariate, Bivariate & Multivariate Analysis in Data Visualisation?
- Using pandas crosstab to create a bar plot
- Exploring Correlation in Python

- Covariance and Correlation
- Factor Analysis | Data Analysis
- Data Mining - Cluster Analysis
- MANOVA Test in R Programming
- MANOVA Test in R Programming
- Python - Central Limit Theorem
- Probability Distribution Function
- Probability Density Estimation & Maximum Likelihood Estimation
- Exponential Distribution in R Programming - dexp(), pexp(), qexp(), and rexp() Functions
- Mathematics | Probability Distributions Set 4 (Binomial Distribution)
- Poisson Distribution | Definition, Formula, Table and Examples
- P-Value: Comprehensive Guide to Understand, Apply, and Interpret
- Z-Score in Statistics | Definition, Formula, Calculation and Uses
- How to Calculate Point Estimates in R?
- Confidence Interval
- Chi-square test in Machine Learning
- Understanding Hypothesis Testing

Data Preprocessing

- ML | Data Preprocessing in Python
- ML | Overview of Data Cleaning
- ML | Handling Missing Values
- Detect and Remove the Outliers using Python

Data Transformation

- Data Normalization Machine Learning
- Sampling distribution Using Python

Time Series Data Analysis

- Data Mining - Time-Series, Symbolic and Biological Sequences Data
- Basic DateTime Operations in Python
- Time Series Analysis & Visualization in Python
- How to deal with missing values in a Timeseries in Python?
- How to calculate MOVING AVERAGE in a Pandas DataFrame?
- What is a trend in time series?
- How to Perform an Augmented Dickey-Fuller Test in R
- AutoCorrelation

Case Studies and Projects

- Top 8 Free Dataset Sources to Use for Data Science Projects
- Step by Step Predictive Analysis - Machine Learning
- 6 Tips for Creating Effective Data Visualizations

Introduction to TensorFlow

-
-
-

Introduction to TensorFlow

TensorFlow is an open-source machine learning library developed by Google. TensorFlow is used to build and train deep learning models as it facilitates the creation of computational graphs and efficient execution on various hardware platforms. The article provides an comprehensive overview of tensorflow.

Introduction to TensorFlow

Table of Content

Introduction to TensorFlow

- TensorFlow
- How to install TensorFlow?
- The Computational Graph
- Variables
- Placeholders
- Linear Regression model using TensorFlow
- `tf.contrib.learn`
- What are TensorFlow APIs?

TensorFlow

TensorFlow is basically a software library for numerical computation using data flow graphs where:

TensorFlow

- nodes in the graph represent mathematical operations.
- edges in the graph represent the multidimensional data arrays (called tensors) communicated between them. (Please note that tensor is the central unit of data in TensorFlow).

TensorFlow

Consider the diagram given below: Here, `add` is a node which represents addition operation. `a` and `b` are input tensors and `c` is the resultant tensor. This flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API!

How to install TensorFlow?

An easy-to-follow guide for TensorFlow installation is available.

How to install TensorFlow?

Here you can refer to this article ? [How to Install Python Tensorflow in Windows?](#)

How to install TensorFlow?

Once you can ensure a successful installation by running this command in python interpreter:

Computational Graph

AnyTensorFlow Coreprogram can be divided into two discrete sections:

Computational Graph

- Building the computational graph. Acomputational graphis nothing but a series of TensorFlow operations arranged into a graph of nodes.
- Running the computational graph. To actually evaluate the nodes, we must run the computational graph within a session. A session encapsulates the control and state of the TensorFlow runtime.

Computational Graph

Now, let us write our very firstTensorFlowprogram to understand above concept:

C++

C++

C++

C++

```
#include <iostream>using namespace std;int main() {cout <<"GFG!";return 0;}
```

Python

Python

Python

Python

```
# importing tensorflowimport tensorflow as tf# creating nodes in computation graphnode1=tf.constant(3, dtype=tf.int32)node2=tf.constant(5, dtype=tf.int32)node3=tf.add(node1,
```

```
node2)# create tensorflow session objectsess=tf.compat.v1.Session()# evaluating node3 and
printing the resultprint("sum of node1 and node2 is :",sess.run(node3))# closing the
sessionsess.close()
```

Python

Output:

Python

Let us try to understand above code:

Python

- Step 1 : Create a computational graphBy creating computational graph, we mean defining the nodes. Tensorflow provides different types of nodes for a variety of tasks. Each node takes zero or more tensors as inputs and produces a tensor as an output.In above program, the nodesnode1andnode2are oftf.constanttype. Aconstantnode takes no inputs, and it outputs a value it stores internally. Note that we can also specify the data type of output tensor usingdtypeargument.
- In above program, the nodesnode1andnode2are oftf.constanttype. Aconstantnode takes no inputs, and it outputs a value it stores internally. Note that we can also specify the data type of output tensor usingdtypeargument.

Python

- In above program, the nodesnode1andnode2are oftf.constanttype. Aconstantnode takes no inputs, and it outputs a value it stores internally. Note that we can also specify the data type of output tensor usingdtypeargument.

Python

- node3is oftf.addtype. It takes two tensors as input and returns their sum as output tensor.

Python

- Step 2 : Run the computational graphIn order to run the computational graph, we need to create

session. To create a session, we simply do:

Python

- Now, we can invoke the `run` method of session object to perform computations on any node:

Python

- Here, `node3` gets evaluated which further invokes `node1` and `node2`. Finally, we close the session using:

Python

Note: Another (and better) method of working with sessions is to use `with` block like this:

Python

The benefit of this approach is that you do not need to close the session explicitly as it gets automatically closed once control goes out of the scope of `with` block.

Variables

TensorFlow has `Variable` nodes too which can hold variable data. They are mainly used to hold and update parameters of a training model. Variables are in-memory buffers containing tensors. They must be explicitly initialized and can be saved to disk during and after training. You can later restore saved values to exercise or analyze the model. An important difference to note between a constant and a `Variable` is:

Variables

A constant's value is stored in the graph and its value is replicated wherever the graph is loaded. A variable is stored separately, and may live on a parameter server.

Variables

Given below is an example using `Variable`:

Python

Python

Python

Python

```
# importing tensorflowimporttensorflow as tf# creating nodes in computation graphnode=tf.Variable(tf.zeros([2,2]))# running computation graphwith tf.Session() as sess:# initialize all global variablessess.run(tf.global_variables_initializer())# evaluating nodeprint("Tensor value before addition:\n",sess.run(node))# elementwise addition to tensornode=node.assign(node+tf.ones([2,2]))# evaluate node againprint("Tensor value after addition:\n", sess.run(node))
```

Python

Output:

Python

In above program:

Python

- We define a node of typeVariableand assign it some initial value.

Python

- To initialize the variable node in current session?s scope, we do:

Python

- To assign a new value to a variable node, we can useassignmethod like this:

Placeholders

A graph can be parameterized to accept external inputs, known asplaceholders. A placeholder is a promise to provide a value later. While evaluating the graph involvingplaceholdernodes, afeed_dictparameter is passed to the session?srunmethod to specify Tensors that provide concrete

values to these placeholders. Consider the example given below:

Python

Python

Python

Python

```
# importing tensorflowimporttensorflow as tf# creating nodes in computation grapha=tf.placeholder(tf.int32, shape=(3,1))b=tf.placeholder(tf.int32, shape=(1,3))c=tf.matmul(a,b)# running computation graphwith tf.Session() as sess:print(sess.run(c, feed_dict={a:[[3],[2],[1]], b:[[1,2,3]]}))
```

Python

Output:

Python

Let us try to understand above program:

Python

- We define placeholder nodes `a` and `b` like this:

Python

- The first argument is the data type of the tensor and one of the optional argument is shape of the tensor.
- We define another node `c` which does the operation of matrix multiplication (`matmul`). We pass the two placeholder nodes as argument.

Python

- Finally, when we run the session, we pass the value of placeholder nodes in `feed_dict` argument

ofsess.run:

Python

- Consider the diagrams shown below to clear the concept:
- Initially:
- After sess.run:

Linear Regression model using TensorFlow

Given below is an implementation of a Linear Regression model using TensorFlow Core API.

Python

Python

Python

Python

```
# importing the dependencies
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Model Parameters
learning_rate=0.01
training_epochs=2000
display_step=200

# Training Data
train_X=np.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,7.042,10.791,5.313,7.9
97,5.654,9.27,3.1])
train_y=np.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.221,2.827
,3.465,1.65,2.904,2.42,2.94,1.3])
n_samples=train_X.shape[0]

# Test Data
test_X=np.asarray([6.83,4.668,8.9,7.91,5.7,8.7,3.1,2.1])
test_y=np.asarray([1.84,2.273,3.2,2.83
1,2.92,3.24,1.35,1.03])

# Set placeholders for feature and target vectors
X=tf.placeholder(tf.float32)
y=tf.placeholder(tf.float32)

# Set model weights and bias
W=tf.Variable(np.random.randn(), name="weight")
b=tf.Variable(np.random.randn(), name="bias")

# Construct a linear model
linear_model=W*X+b

# Mean squared error
cost=tf.reduce_sum(tf.square(linear_model-y))/(2*n_samples)

# Gradient descent
optimizer=tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

# Initializing the variables
init=tf.global_variables_initializer()

# Launch the graph with tf.Session() as sess:
# Load
```

```

initialized variables in current session sess.run(init) # Fit all training
data for epoch in range(training_epochs): # perform gradient descent steps
    sess.run(optimizer, feed_dict={X: train_X, y: train_y}) # Display logs per epoch
    step if (epoch+1)%display_step==0: c=sess.run(cost, feed_dict={X: train_X, y:
    train_y}) print("Epoch:{0:6} \t Cost:{1:10.4} \t W:{2:6.4} \t b:{3:6.4}".format(epoch+1, c, sess.run(W),
    sess.run(b))) # Print final parameter values
    print("Optimization Finished!") training_cost=sess.run(cost, feed_dict={X: train_X, y: train_y})
    print("Final training cost:", training_cost, "W:", sess.run(W), "b:", sess.run(b), '\n')
    # Graphic display
    plt.plot(train_X, train_y, 'ro', label='Original data')
    plt.plot(train_X, sess.run(W)*train_X+sess.run(b), label='Fitted line')
    plt.legend()
    plt.show() # Testing the model
    testing_cost=sess.run(tf.reduce_sum(tf.square(linear_model-y))/(2*test_X.shape[0]), feed_dict={X: test_X, y: test_y})
    print("Final testing cost:", testing_cost)
    print("Absolute mean square loss difference:", abs(training_cost-testing_cost))
    # Display fitted line on test data
    plt.plot(test_X, test_y, 'bo', label='Testing data')
    plt.plot(train_X, sess.run(W)*train_X+sess.run(b), label='Fitted line')
    plt.legend()
    plt.show()

```

Python

Output:

Python

Let us try to understand the above code.

Python

- First of all, we define some parameters for training our model, like:

Python

- Then we define placeholder nodes for feature and target vector.

Python

- Then, we define variable nodes for weight and bias.

Python

- `linear_model` is an operational node which calculates the hypothesis for the linear regression model.

Python

- Loss (or cost) per gradient descent is calculated as the mean squared error and its node is defined as:

Python

- Finally, we have the `optimizer` node which implements the Gradient Descent Algorithm.

Python

- Now, the training data is fit into the linear model by applying the Gradient Descent Algorithm. The task is repeated `training_epochs` number of times. In each epoch, we perform the gradient descent step like this:

Python

- After every `display_step` number of epochs, we print the value of current loss which is found using:

Python

- The model is evaluated on test data and `testing_cost` is calculated using:

tf.contrib.learn

`tf.contrib.learn` is a high-level TensorFlow library that simplifies the mechanics of machine learning, including the following:

tf.contrib.learn

- running training loops
- running evaluation loops
- managing data sets

- managing feeding

tf.contrib.learn

Let us try to see the implementation of Linear regression on same data we used above using `tf.contrib.learn`.

Python

Python

Python

Python

```
# importing the dependencies
import tensorflow as tf
import numpy as np

# declaring list of features
features = [tf.contrib.layers.real_valued_column("X")]

# creating a linear regression estimator
estimator = tf.contrib.learn.LinearRegressor(feature_columns=features)

# training and test data
train_X = np.asarray([3.3, 4.4, 5.5, 6.71, 6.93, 4.168, 9.779, 6.182, 7.59, 2.167, 7.042, 10.791, 5.313, 7.997, 5.654, 9.27, 3.1])
train_y = np.asarray([1.7, 2.76, 2.09, 3.19, 1.694, 1.573, 3.366, 2.596, 2.53, 1.221, 2.827, 3.465, 1.65, 2.904, 2.42, 2.94, 1.3])
test_X = np.asarray([6.83, 4.668, 8.9, 7.91, 5.7, 8.7, 3.1, 2.1])
test_y = np.asarray([1.84, 2.273, 3.2, 2.831, 2.92, 3.24, 1.35, 1.03])

# function to feed dict of numpy arrays into the model for training
input_fn = tf.contrib.learn.io.numpy_input_fn({"X": train_X}, train_y, batch_size=4, num_epochs=2000)

# function to feed dict of numpy arrays into the model for testing
test_input_fn = tf.contrib.learn.io.numpy_input_fn({"X": test_X}, test_y)

# fit training data into estimator
estimator.fit(input_fn=input_fn)

# print value of weight and bias
W = estimator.get_variable_value('linear/X/weight')[0][0]
b = estimator.get_variable_value('linear/bias_weight')[0]

print("W:", W, "\tb:", b)

# evaluating the final loss
train_loss = estimator.evaluate(input_fn=input_fn)['loss']
test_loss = estimator.evaluate(input_fn=test_input_fn)['loss']

print("Final training loss:", train_loss)
print("Final testing loss:", test_loss)
```

Python

Output:

Python

Let us try to understand the above code.

Python

- The shape and type of feature matrix is declared using a list. Each element of the list defines the structure of a column. In above example, we have only 1 feature which stores real values and has been given a nameX.

Python

- Then, we need an estimator. An estimator is nothing but a pre-defined model with many useful methods and parameters. In above example, we use a Linear Regression model estimator.

Python

- For training purpose, we need to use an input function which is responsible for feeding data to estimator while training. It takes the feature column values as dictionary. Many other parameters like batch size, number of epochs, etc can be specified.

Python

- To fit training data to estimator, we simply use `fit` method of estimator in which input function is passed as an argument.

Python

- Once training is complete, we can get the value of different variables using `get_variable_value` method of estimator. You can get a list of all variables using `get_variable_names` method.

Python

- The mean squared error/loss can be computed as:

What are TensorFlow APIs?

TensorFlow provides multiple APIs (Application Programming Interfaces). These can be classified into 2 major categories:

Low level API:

- complete programming control
- recommended for machine learning researchers.
- provides fine levels of control over the models.
- TensorFlow Core is the low-level API of TensorFlow.

High level API:

- built on top of TensorFlow Core
- easier to learn and use than TensorFlow Core
- make repetitive tasks easier and more consistent between different users.
- `tf.contrib.learn` is an example of a high-level API.

High level API:

High level API:

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the Three 90 Challenge? complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

High level API:

- AI-ML-DS
- Python
- Technical Scripter
- Tensorflow

High level API:

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy

- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- What is Google Cloud Platform (GCP)?
- Introduction to Google Cloud Platform
- Cloud Storage in Google Cloud Platform (GCP)
- Features of Google Cloud Platform
- Google Cloud Platform - Introduction to Qwiklabs

Compute Services

- Google Cloud Platform - Compute Services
- Cloud Functions in GCP
- How to Use Google Cloud Function with Python ?
- Difference Between Google Cloud Compute Engine and App Engine
- Google Cloud Platform - Automatic Vs User-Managed Replication Policy

Storage and Database Services

- Google Cloud Platform - Cloud Storage
- Google File System
- Introduction to Google Cloud Bigtable

Networking Services

- Google Cloud Platform Networking Services

Security Services

- Google Cloud Platform Security
- Access Control for Disaster Avoidance in Google Cloud IoT Core using IAM Policy

Data Integration and Analytics Services

- Introduction to Databricks
- Google Cloud Platform - Introduction to BigQuery
- Google Cloud Platform - Introduction to BigQuery Sandbox
- Google Cloud Platform - Tables in BigQuery
- Google Cloud Platform- BigQuery(Running Queries, advantage and disadvantage)
- Google Cloud Platform - User Defined Functions in BigQuery
- Google Cloud Platform - Working with External Data in BigQuery
- Google Cloud Platform - Loading Data to BigQuery
- Google Cloud Platform - Implementing Authorized View in BigQuery
- Google Cloud Platform - Query History vs Saved Query vs Shared Query in BigQuery

- Google Cloud Platform - Managing Access using IAM in BigQuery
- Google Cloud Platform - Data Visualization in BigQuery
- Google Cloud Platform - Data Security in BigQuery

Management tools and Monitoring Services

- Google Cloud Platform - High Level Overview of Migrate for Anthos

GCP DevOps

- Google Cloud Platform - Using Config Sync for Managing Kubernetes
- Google Cloud Platform - Deploying Django & its Content Management Systems

Miscellaneous

- Difference Between Google Cloud and AWS
- How To Share File From Host Machine(Windows) To Guest Machine(Linux)
- Deployment Models in OpenStack
- How to Build G Suite Add-ons with Google Apps script?
- Google Cloud Platform - Introduction to PhoneInfoga an OSINT Reconnaissance Tool
- Generating API Keys For Using Any Google APIs
- Google Cloud Platform - Understanding Federated Learning on Cloud

Google Cloud Platform Tutorial

-
-
-

Google Cloud Platform Tutorial

Google Cloud Platform is a set of cloud computing services provided by Google that allow you to store, manage, and analyze data. It is also used for developing, deploying, and scaling applications on Google's environment. It was announced as the leading cloud platform in Gartner's IaaS Magic

Quadrant in 2018.

Google Cloud Platform Tutorial

Likewise, it provides tools for developers to build and deploy applications securely and reliably, with access to advanced security features. Services of Google Cloud Platform can be accessed by developers, cloud administrators, and other enterprise IT professionals through the public internet or a dedicated network connection.

Google Cloud Platform Tutorial

Google Cloud Platform Tutorial

In this Google Cloud Platform tutorial you'll learn all the basic to advanced concepts like Google Cloud, Google Cloud Storage, Google Cloud console, google cloud services, google cloud servers, google cloud hosting, etc. This tutorial is designed to help beginners and professionals.

Google Cloud Platform Tutorial

The tutorial provides learners with a complete guide to using the Google Cloud platform as a service. Whether you are new to cloud computing or looking to expand your knowledge and skills, this tutorial will provide you with the necessary information and resources to effectively use GCP.

What is Cloud Computing?

Cloud computing is the distribution of computing resources on demand via the internet, including servers, storage, databases, software, and more. It lowers costs and boosts efficiency by enabling individuals and organizations to access and utilize these resources without requiring on-site infrastructure or upkeep.

What is Cloud Computing?

Due to its improved manageability, quick speed, and low maintenance requirements, cloud computing is advancing. It aids businesses in reducing their use of resources and total infrastructure expenditures. Instead of keeping data on a local server or their devices, users can save, change, or

process it on the internet or a cloud-based platform with the help of a remote server.

Introduction

In this section we will discussed all about GCP and its features, so explore this Google Cloud Platform introductory section and get all details of GCP.

Introduction

- What is Google Cloud Platform (GCP)?
- Introduction to Google Cloud Platform
- Cloud Storage in Google Cloud Platform (GCP)
- Features of GCP
- Google Cloud Platform ? Introduction to Qwiklabs

Compute Services

- Key GCP Compute Services
- Google Compute Engine
- Cloud Functions in GCP
- How to Use Google Cloud Function with Python?
- Difference Between Google Cloud Compute Engine and App Engine
- Google Cloud Platform ? Automatic Vs User-Managed Replication Policy

Storage and Database Services

- Google Cloud Storage
- Google File System
- Introduction to Google Cloud Bigtable

Networking Services

- Google Cloud Platform Networking Services

Security Services

- What is GCP(Google Cloud Platform) Security?
- Access Control for Disaster Avoidance in Google Cloud IoT Core using IAM Policy

Data Integration and Analytics Services

- Introduction to Databricks
- Google Cloud Platform ? Introduction to BigQuery
- Google Cloud Platform ? Introduction to BigQuery Sandbox
- Google Cloud Platform ? Tables in BigQuery
- Google Cloud Platform ? Running Queries in BigQuery
- Google Cloud Platform ? User-Defined Functions in BigQuery
- Google Cloud Platform ? Working with External Data in BigQuery
- Google Cloud Platform ? Loading Data to BigQuery
- Google Cloud Platform ? Implementing Authorized View in BigQuery
- Google Cloud Platform ? Query History vs Saved Query vs Shared Query in BigQuery
- Google Cloud Platform ? Managing Access using IAM in BigQuery
- Google Cloud Platform ? Data Visualization in BigQuery
- Google Cloud Platform ? Data Security in BigQuery

Management tools and Monitoring Services

- Google Cloud Platform ? High-Level Overview of Migrate for Anthos

GCP DevOps

- Google Cloud Platform ? Using Config Sync for Managing Kubernetes
- Google Cloud Platform ? Deploying Django & its Content Management Systems

Miscellaneous

- Difference Between Google Cloud and AWS
- How To Share Files From Host Machine(Windows) To Guest Machine(Linux)
- Deployment Models in OpenStack

- How to Build G Suite Add-ons with Google Apps script?
- Google Cloud Platform ? Introduction to PhoneInfoga an OSINT Reconnaissance Tool
- Generating API Keys For Using Any Google APIs
- Google Cloud Platform ? Understanding Federated Learning on Cloud

Why we use Google Cloud Platform?

Google Cloud Platform is one of the famous cloud providers in the IT industry. Google provides a scalable and reliable platform for building, testing, and deploying applications in a real-time environment.

Features of Google Cloud Platform

- Affordability: The price model of GCP is awesome it allows you to pay only for what you use.
- Speed: Reduce the serving cost and Accelerate content delivery to users through Google Cloud CDN.
- Consulting Services: To build your cloud strategies google provides its own Google experts.
- Cloud Data Transfer: By simple drag-and-drop method you can transfer your data to Google Cloud easily.
- Admin Control: Create or remove users? accounts, and add additional security options.
- Fast Development: GCP provides some tools and libraries which make development on GCP faster.

Conclusion

Google Cloud Platform (GCP) is a powerful and flexible cloud computing platform that offers a wide range of services for businesses and individuals. This tutorial provided an overview of GCP and its key features, as well as a detailed look at some of the most popular services offered by the platform. Both beginners and experts will easily understand this tutorial. With the ever-growing demand for cloud computing services, GCP is a valuable addition to any IT professional or developer's skill set.

How can I get started with Google Cloud Platform?

To get started with Google Cloud Platform, you can create a GCP account, explore the documentation and tutorials available on the official GCP website, and consider taking online courses or certifications to enhance your skills.

What are the key features of Google Cloud Platform?

Some key features of Google Cloud Platform include scalable and flexible infrastructure, data analytics and machine learning capabilities, robust security measures, and a global network of data centers for high performance and low latency.

How does Google Cloud Platform differ from other cloud providers?

Google Cloud Platform sets itself apart from other cloud providers with its emphasis on industry-leading data analytics, machine learning, and artificial intelligence capabilities, as well as its global network infrastructure and commitment to sustainability.

What are some popular use cases for Google Cloud Platform?

Popular use cases for Google Cloud Platform include application and website hosting, data storage and analysis, IoT solutions, machine learning and AI applications, and enterprise-level business solutions.

What are some popular use cases for Google Cloud Platform?

What are some popular use cases for Google Cloud Platform?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

What are some popular use cases for Google Cloud Platform?

- DevOps
- Google Cloud Platform
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation

- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp

- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux

- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures

- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- Introduction to Microsoft Azure | A Cloud Computing Service
- Microsoft Azure - High-Level Overview of Architecture Center
- How Microsoft Azure Works?
- Microsoft Azure - Using the Azure Quickstart Centre
- Microsoft Azure - Getting Free Trial with Azure Quickstart Center

Storage

- Microsoft Azure - Storage Accounts
- Microsoft Azure - Storage in Azure Cloud Shell
- Microsoft Azure - Mounting Azure Storage in a Container App
- Microsoft Azure - Concept of Blob Versioning in Azure Storage
- Microsoft Azure - Using Blob Versioning
- Microsoft Azure - Automatically Manage Azure Blobs Lifecycles
- Microsoft Azure - Add Storage Blob Owner Role Assignment For Azure AD User
- Microsoft Azure- Disk Storage in Microsoft Azure
- How to Setup a Azure Storage Account For Data Archive?
- Microsoft Azure - Hosting a Static Website with Azure Storage
- Microsoft Azure - Azure Managed Disk State Details using PowerShell
- Microsoft Azure - Mount Azure Storage in Container App In App Service
- Microsoft Azure - Delete Orphaned Disks using PowerShell Script

Network

- What Are Azure Data Security Features?
- Microsoft Azure - Virtual Network
- Microsoft Azure - Find Orphaned Network Interface Cards(NICs)
- What is Microsoft Azure Network Watcher?
- Microsoft Azure - Finding the Right Load Balancing Service
- Microsoft Azure - Application Security with Azure Active Directory
- Microsoft Azure - Find and Delete Orphaned Public IP addresses in Azure Portal
- Microsoft Azure - Protecting Hybrid Cloud Workloads using Azure Defender

Compute Virtual Machines

- What are Azure Virtual Machines?
- Microsoft Azure - Introduction to Azure VM Extension

- Microsoft Azure - Simpler Management of Virtual Machine
- Microsoft Azure - Managing Multiple Virtual Machines
- Create Free Windows Virtual Machine in Azure
- Microsoft Azure - Pre-built Azure VMs
- Microsoft Azure - Archive Azure VM to Azure Storage Account
- How to Reset a Azure VM Password?
- Microsoft Azure - Improvements in Linux Virtual Machine
- Microsoft Azure - Connecting Linux VM using SSH
- Microsoft Azure - Connecting Windows VM using RDP
- Microsoft Azure - Azure VM Disk Space in Percentage using Azure KQL
- Microsoft Azure - Creating an OS Disk Snapshot of a Azure VM
- Microsoft Azure - Reduce Cost Using Virtual Machine
- Microsoft Azure - Resize a Azure Virtual Machine for Windows or Linux Servers
- Microsoft Azure - Resizing Virtual Machine Using PowerShell Script
- Microsoft Azure - Configuring Backup For Azure App Services
- Microsoft Azure - Getting Started With Azure Backup For Protecting an IaaS VM
- Defense in Depth Strategy in Microsoft Azure
- Microsoft Azure - Enable IIS Logs for Monitoring
- Microsoft Azure - Using Azure Monitor Workbooks
- Microsoft Azure - Enable Azure Monitor VM Insights Agent For Azure VM
- Microsoft Azure - Introduction to Spot Virtual Machines
- Azure Virtual Machine for Machine Learning
- Microsoft Azure - Tracking Memory Utilization of Azure VM using KQL Log Query
- Microsoft Azure - Track CPU Utilization of a Azure Virtual Machines using KQL Log Query
- Microsoft Azure - Heartbeat KQL Log Monitoring Query
- Microsoft Azure - Firewall Network Flow Logs with TimeGenerated using KQL
- Microsoft Azure - Setting up a Auto Shutdown For Azure Virtual Machine

- Microsoft Azure - Get CPU Utilization Data of a Azure VM in Azure Portal
- Microsoft Azure - CPU and Memory Utilization of Azure VMs in a Single Table Chart
- Microsoft Azure - Retrieve Azure Virtual Machine Admin User Name
- Microsoft Azure - Enable Boot Diagnostics For a VM in Azure
- Microsoft Azure - Disable Boot Diagnostics For a VM in Azure
- Microsoft Azure - Setup Auto Startup and Shutdown of an Azure VM
- 7 Tips to Reduce Cost with Azure Virtual Machines
- Microsoft Azure - Assigning Tags to a Azure VM
- Microsoft Azure - Check Virtual Machine Creation Date and Time
- Microsoft Azure - VMs Patch Update Summary by Installation Status
- Microsoft Azure - Graph Query to Get Properties of Azure VM Resource
- Microsoft Azure - Check for Apache Log4J Vulnerability in Azure VMs
- Microsoft Azure - Get Azure VM Properties using Azure PowerShell
- Microsoft Azure - Manage Azure VMs using Azure PowerShell Commands
- Microsoft Azure - Check Status of Azure VM using Azure PowerShell
- Microsoft Azure - Azure CLI Commands to Manage Azure VMs
- Microsoft Azure - KQL Query to Get the VM Computer Properties
- Microsoft Azure - Create SHH Key to Manage Azure Linux VMs
- Microsoft Azure - VM CPU Utilisation Across Subscriptions with Range
- Microsoft Azure - Common Virtual Machine Issues
- Microsoft Azure - Manage Virtual Machine via Azure Mobile App
- Microsoft Azure - Increase Processors & Memory Size of Windows Virtual Machine
- Microsoft Azure - Troubleshoot Azure VM Inbound and Outbound Connections
- Microsoft Azure - Deletion of Snapshots using PowerShell Script
- Microsoft Azure - Connect to a Storage Account using Private Link
- Microsoft Azure - Accessing Virtual Machines using Bastion
- Microsoft Azure - Getting VMComputer Records Properties

Compute Functions

- Microsoft Azure - Monitoring Azure Functions
- Microsoft Azure - Creating Azure Functions From VS Code
- Microsoft Azure - Debugging Azure Functions using VS Code
- Microsoft Azure - Deploy Azure Functions using VS Code
- Microsoft Azure - Renaming an Azure Functions
- Microsoft Azure - Simple way to Create a Function App
- Microsoft Azure - New Features in Azure Functions
- Microsoft Azure - Using VS Code for Azure Function Projects
- Microsoft Azure - Custom Handlers for Azure Functions

Compute Monitoring

- Microsoft Azure - Enable Windows Performance Counters in Azure for Monitoring
- Microsoft Azure - Enable Linux Performance Counters in Azure for Monitoring
- Microsoft Azure - Enable Windows Event Logs in Azure for Monitoring
- Microsoft Azure - Enable Linux System Logs in Azure for Monitoring

Azure Services

- Microsoft Azure - Introduction to Quickstart Center
- Microsoft Azure- Add Extensions in Azure App Service
- Microsoft Azure - Working with App Service Domains
- Microsoft Azure - Create Web Apps in Azure
- Microsoft Azure - Creating an App Service Plan
- Microsoft Azure - Creating a Logic App of Consumption Type
- Microsoft Azure - Azure App Service Managed Certificates
- Microsoft Azure - Deploy a Static Web App
- Microsoft Azure - Removing Unused Services From Azure
- What is Microsoft Azure Functions?

- Microsoft Azure - RDP to Azure Virtual Machines using Azure Bastion
- Microsoft Azure - Getting started with Azure Static Web Apps
- Microsoft Azure - Cloning Web Apps using Azure App Services
- Microsoft Azure - Enable Local Cache on Azure App
- Microsoft Azure - Exploring the Azure Mobile App Home Screen
- Microsoft Azure - Container Registration for App Deployment
- Microsoft Azure - Roll Back Versions of Azure Logic Apps
- Microsoft Azure - Modifying API Connections With Logic Apps
- Microsoft Azure - Azure Media Service
- Microsoft Azure - Using Github Action in Azure App Service
- Microsoft Azure - Using JSON with Azure Logic Apps
- Microsoft Azure - Configuring Log Analytics Data Retention
- Microsoft Azure - Creating JSON Snippet For Policy Initiative Creation using CSV File
- Microsoft Azure - Working with Files in Azure App Service
- Microsoft Azure - Using Power Apps in VS Code

Resources

- Microsoft Azure - Preventing Changes to Resources in Azure App Services
- Microsoft Azure- Writing Queries and Creating Dashboards using Azure Resource Graph
- Microsoft Azure - Enabling Resource Providers for Azure Subscription
- Microsoft Azure - Find Orphaned Disks
- Microsoft Azure - Handling Orphaned NSGs
- Microsoft Azure - Configuring an Alert Rule with Azure Monitor
- Microsoft Azure - Count of Azure Resources using Resource Graph Query
- Microsoft Azure - Create Alert For Conditional Access Policy Changes
- Microsoft Azure - Create Pie Charts For Orphaned Resources using KQL
- Microsoft Azure - Patch Management Update Summary Status

- Microsoft Azure - Check Resource Owner in Azure using KQL
- Microsoft Azure - Check Enabled and Disabled Diagnostics for Azure Resources
- Microsoft Azure - Azure Firewall Flow Logs From Select Source IP
- Microsoft Azure - Resource Tagging and Best Practices
- Microsoft Azure - Using the Azure Activity Log
- Microsoft Azure - Scaling Inventory Resources
- Microsoft Azure - Moving Resources Among Groups

Cognitive Services

- Build a Text Translator Web App using Flask and Azure Cognitive Services
- Microsoft Azure - Introduction to Metric Advisor
- Microsoft Azure - Cognitive Service Text Analytics in Web App Containers
- Microsoft Azure - Opinion Mining with Azure Cognitive Services

Dashboards

- Microsoft Azure - Customizing Charts on Azure Dashboard
- Microsoft Azure - Customizing Azure Dashboard Tiles
- Microsoft Azure - Auto Refreshing Dashboards
- Microsoft Azure - Pinning to Portal Dashboard

Databases

- Microsoft Azure - Database Availability and Consistency For Azure SQL
- Microsoft Azure SQL Database | Complete Tutorial
- Microsoft Azure - Security in Azure SQL
- Microsoft Azure - Identifying & Accessing Management for Azure SQL
- Microsoft Azure - Security Capabilities and Tasks of Azure SQL
- Microsoft Azure - Configure Azure SQL with Azure CLI
- Microsoft Azure - Configure Auditing for Azure SQL Database

- Microsoft Azure - Azure SQL Managed Instance Deployment Options
- Microsoft Azure - Azure SQL Managed Instance
- Microsoft Azure - PaaS vs. IaaS in Azure SQL
- Microsoft Azure - Management Interfaces for Azure SQL
- Microsoft Azure - Opportunities in the Cloud with Azure SQL
- Microsoft Azure - Create Azure SQL Database
- Microsoft Azure - Introduction to Azure Database For MySQL
- Microsoft SQL Server Active Directory Authentication on Linux Machine
- Introduction to Azure Cosmos DB
- Microsoft Azure - Notebooks in Cosmos DB
- Microsoft Azure - Using C# in CosmoDB
- Microsoft Azure - Choosing a Partition Key in Cosmos DB
- Microsoft Azure - Introduction to Azure Data Factory
- Microsoft Azure - Migrating From MySQL to Cloud
- Microsoft Azure - Advanced Threat Protection for Azure SQL Database
- Microsoft Azure - Using Flexible Server in Azure Database for PostgreSQL
- Microsoft Azure - Accelerating Cloud Migration & Datacenter Discovery

DevOps

- Azure DevOps: An Introduction, Services, Projects, Benefits, Drawbacks
- Microsoft Azure - Availability Zones For Kubernetes Cluster
- Microsoft Azure - Zone Redundancy and SLA of Azure SQL
- Microsoft Azure - Create Project in Azure Devops using Basic Process
- Microsoft Azure ? Create Project in Azure Devops using Agile Process
- Microsoft Azure - Starting & Stopping a Azure Kubernetes Service Cluster
- Microsoft Azure - Introduction to Azure Kung Fu Toolkit
- Microsoft Azure - Introduction to Kubernetes Diagnostics

- How to Create Backlog and Sprint in Agile Process?
- Microsoft Azure - Running an App inside a Docker Container Image
- How to Push a Container Image to a Docker Repository?
- Microsoft Azure - Manage and Export Azure Policies with GitHub Integration
- Microsoft Azure Arc Enabled Kubernetes
- Microsoft Azure - Introduction to Azure Arc
- Microsoft Azure - Deploying Web Applications Using GitHub Actions
- Microsoft Azure - Messaging with Azure Web PubSub
- Microsoft Azure - Testing in Production Sites Feature using PowerShell
- Microsoft Azure - Build and Deploy App with Azure SDK For Java

Machine Learning and IoT

- Microsoft Azure - Introduction to Azure Sentinel
- Machine Learning with Microsoft Azure ML Studio Without Code
- What is the Best Azure IoT Service for You?
- Microsoft Azure - Key IoT Products
- Microsoft Azure - Using Azure Form Recognizer

Miscellaneous

- Microsoft Azure - Productivity Tips For Azure portal
- Microsoft Azure - Get Azure Policies Definitions and Policy Initiatives Definitions
- Microsoft Azure - Create a Custom Policy Initiative in Azure
- Technical Overview of Microsoft Mesh
- Microsoft Azure - Working with Azure Health Bot
- Microsoft Azure - Introduction to Azure Digital Twins
- Microsoft Azure - Working with Themes in Azure
- Microsoft Azure - Understanding Azure KeyBoard Shortcuts
- Microsoft Azure - Using Table Parameter in Azure CLI

- Microsoft Azure - Add Azure Cloud Shell to VS Code
- Microsoft Azure - Accessing Cloud Shell from Microsoft Docs
- Microsoft Azure - Azure App Servers Routing using PowerShell
- Microsoft Azure - Introduction to SMART
- Microsoft Azure - Lift & Shift Approach for Infrastructure Migration
- Microsoft Azure - Configure Diagnostic Settings For Azure Subscription
- Microsoft Azure - Checking your Access on Azure Subscription
- Microsoft Azure - Enabling Logs for Troubleshooting the Azure Firewall Rules
- Microsoft Azure - Rebooting an Application Gateway
- Microsoft Azure - PAL Tagging in Azure
- Microsoft Azure - Using Cost Management in Azure Portal
- Microsoft Azure - View Delegated Roles in Azure Lighthouse
- Microsoft Azure - Using the Global Search
- Microsoft Azure - Quick Copy & Paste on Portal
- Microsoft Azure - Alerts as Notifications on Phone
- Microsoft Azure - Developing ARM Templates using Azure Bicep

Microsoft Azure Tutorial

-
-
-

Microsoft Azure Tutorial

Designed by Microsoft in 2010, Microsoft Azure is one of the widely used cloud computing platforms. Azure provides a wide variety of services, such as cloud storage, compute services, network services, cognitive services, databases, analytics, and IoT. It makes building, deploying, and managing applications very easy. All the Microsoft Azure fundamentals are also described to better understand readers.

Microsoft Azure Tutorial

Microsoft Azure Tutorial

Microsoft Azure tutorial is designed for beginners to learn about Microsoft Azure and its various cloud computing services. It covers important concepts and provides a comprehensive understanding of Azure's services. The tutorial includes a variety of topics that delve into the uses and services of Azure in depth.

Prerequisites to Learn Azure

To learn Microsoft Azure there is no need of any extra knowledge any one can easily learn the Azure if they have the below listed set of skills.

Prerequisites to Learn Azure

- Basic understanding of Azure concept.
- Understanding of Cloud Concepts
- Understanding if basic infrastructure management, database management, and software development.

Introduction of Microsoft Azure

Discover how Azure empowers businesses to build, deploy, and manage applications across Microsoft's global network of data centers, supporting a wide array of programming languages, frameworks, and tools. Whether you're new to cloud computing or looking to expand your expertise, Azure provides the tools and scalability needed to meet diverse business needs effectively.

Introduction of Microsoft Azure

- Introduction to Microsoft Azure
- Azure Architecture
- How Azure Works?
- Microsoft Azure ? Using the Azure Quickstart Centre

- Microsoft Azure ? Getting a Free Trial with Azure Quickstart Center

Storage

Here in this section, you will discover how Azure Storage provides durability, scalability, and high availability, making it suitable for a wide array of use cases from simple file sharing to complex big data analytics and disaster recovery solutions.

Storage

- Azure Storage Account
- Storage in Azure Cloud Shell
- Azure Storage in Container App
- Blob versioning
- Microsoft Azure ? Using Blob Versioning
- Manage Blob Lifecycle
- Microsoft Azure ? Automatically Manage Azure Blobs Lifecycles
- Microsoft Azure ? Add Storage Blob Owner Role Assignment For Azure AD User
- Azure Disk Storage
- How to Setup an Azure Storage Account For Data Archive?
- IAM blob storage
- Host a Static Website using Storage
- Microsoft Azure- Disk Storage in Microsoft Azure
- Microsoft Azure ? Azure Managed Disk State Details using PowerShell
- Microsoft Azure ? Mount Azure Storage in Container App In App Service
- Microsoft Azure ? Mounting Azure Storage in a Container App
- Microsoft Azure ? Delete Orphaned Disks using PowerShell Script
- Microsoft Azure ? Storage in Azure Cloud Shell

Network

In this section of Microsoft Azure tutorial you will learn all about Azure network like virtual network, network watcher, and more.

Network

- Data Security
- Virtual Network
- Microsoft Azure ? Find Orphaned Network Interface Cards(NICs)
- Network Watcher
- How to select the right load balancer
- Azure Virtual Network
- Microsoft Azure ? Application Security with Azure Active Directory
- Microsoft Azure ? Find and Delete Orphaned Public IP addresses in Azure Portal
- Microsoft Azure ? Protecting Hybrid Cloud Workloads using Azure Defender

A. Virtual Machines

- Azure Virtual Machines
- Microsoft Azure ? Introduction to Azure VM Extension
- Management of Virtual Machines
- Microsoft Azure ? Managing Multiple Virtual Machines
- How to setup windows VM
- What are Pre-built VM
- Microsoft Azure ? Introduction to Spot Virtual Machines
- Microsoft Azure ? Archive Azure VM to Azure Storage Account
- How to Reset an Azure VM Password?
- How to create a Linux virtual machine using SSH keys
- Microsoft Azure ? Connecting Linux VM using SSH
- Microsoft Azure ? Connecting Windows VM using RDP
- Microsoft Azure ? RDP to Azure Virtual Machines using Azure Bastion

- Azure VM Disk Space in Percentage using Azure KQL
- Microsoft Azure ? Creating an OS Disk Snapshot of an Azure VM
- Microsoft Azure ? Reduce Cost Using Virtual Machine
- Microsoft Azure ? Resize an Azure Virtual Machine for Windows or Linux Servers
- Microsoft Azure ? Resizing Virtual Machine Using PowerShell Script
- Microsoft Azure ? Configuring Backup For Azure App Services
- Azure Backup For Protecting an IaaS VM
- Configuring Backup For Azure App Services
- Defense in Depth Strategy in Microsoft Azure
- Microsoft Azure ? Enable IIS Logs for Monitoring
- Using Azure Monitor Workbooks
- Enable Azure Monitor VM Insights Agent
- Introduction to Spot Virtual Machines
- VM for Machine Learning
- Microsoft Azure ? Tracking Memory Utilization of Azure VM using KQL Log Query
- Microsoft Azure ? Track CPU Utilization of Azure Virtual Machines using KQL Log Query
- Heartbeat KQL Log Monitoring Query
- Microsoft Azure ? Firewall Network Flow Logs with TimeGenerated using KQL
- Microsoft Azure ? Setting up an Auto Shutdown For Azure Virtual Machine
- Microsoft Azure ? Get CPU Utilization Data of an Azure VM in the Azure Portal
- Microsoft Azure ? CPU and Memory Utilization of Azure VMs in a Single Table Chart
- Microsoft Azure ? Retrieve Azure Virtual Machine Admin User Name
- Microsoft Azure ? Enable Boot Diagnostics For a VM in Azure
- Microsoft Azure ? Disable Boot Diagnostics For a VM in Azure
- Microsoft Azure ? Setup Auto Startup and Shutdown of an Azure VM
- 7 Tips to Reduce Cost with Azure Virtual Machines
- Microsoft Azure ? Assigning Tags to an Azure VM

- Microsoft Azure ? Check Virtual Machine Creation Date and Time
- Microsoft Azure ? VMs Patch Update Summary by Installation Status
- Microsoft Azure ? Graph Query to Get Properties of Azure VM Resource
- Microsoft Azure ? Enable Azure Monitor VM Insights Agent For Azure VM
- Microsoft Azure ? Check for Apache Log4J Vulnerability in Azure VMs
- Microsoft Azure ? Get Azure VM Properties using Azure PowerShell
- Microsoft Azure ? Manage Azure VMs using Azure PowerShell Commands
- Microsoft Azure ? Check the Status of Azure VM using Azure PowerShell
- Microsoft Azure ? Azure CLI Commands to Manage Azure VMs
- Microsoft Azure ? Azure VM Disk Space in Percentage using Azure KQL
- Microsoft Azure ? KQL Query to Get the VM Computer Properties
- Microsoft Azure ? Create SSH Key to Manage Azure Linux VMs
- Microsoft Azure ? VM CPU Utilisation Across Subscriptions with Range
- Microsoft Azure ? Common Virtual Machine Issues
- Microsoft Azure ? Manage Virtual Machine via Azure Mobile App
- Microsoft Azure ? Simpler Management of Virtual Machine
- Microsoft Azure ? Increase Processors & Memory Size of Windows Virtual Machine
- Microsoft Azure ? Troubleshoot Azure VM Inbound and Outbound Connections
- Microsoft Azure ? Deletion of Snapshots using PowerShell Script
- Microsoft Azure ? Connect to a Storage Account using a Private Link
- Microsoft Azure ? Accessing Virtual Machines using Bastion
- Microsoft Azure ? Getting VMComputer Records Properties
- Microsoft Azure ? Improvements in Linux Virtual Machine

B. Functions

- Monitoring Azure Functions
- Microsoft Azure ? Creating Azure Functions From VS Code

- Microsoft Azure ? Debugging Azure Functions using VS Code
- Microsoft Azure ? Deploy Azure Functions using VS Code
- Microsoft Azure ? Renaming an Azure Functions
- Microsoft Azure ? Simple way to Create a Function App
- Microsoft Azure ? Monitoring Azure Functions
- Microsoft Azure ? New Features in Azure Functions
- Microsoft Azure ? Using VS Code for Azure Function Projects
- Microsoft Azure ? Custom Handlers for Azure Functions

C. Monitoring

- Microsoft Azure ? Enable Windows Performance Counters in Azure for Monitoring
- Microsoft Azure ? Enable Linux Performance Counters in Azure for Monitoring
- Microsoft Azure ? Enable Windows Event Logs in Azure for Monitoring
- Microsoft Azure ? Enable Linux System Logs in Azure for Monitoring
- Microsoft Azure ? Using Azure Monitor Workbooks

Azure Services

- Microsoft Azure ? Introduction to Quickstart Center
- Microsoft Azure- Add Extensions in Azure App Service
- Microsoft Azure ? Working with App Service Domains
- Microsoft Azure ? Create Web Apps in Azure
- Microsoft Azure ? Creating an App Service Plan
- Microsoft Azure ? Creating a Logic App of Consumption Type
- Microsoft Azure ? Azure App Service Managed Certificates
- Microsoft Azure ? Deploy a Static Web App
- Microsoft Azure ? Removing Unused Services From Azure
- Azure Functions
- RDP to Azure Virtual Machines using Azure Bastion

- Microsoft Azure ? Getting started with Azure Static Web Apps
- Microsoft Azure ? Cloning Web Apps using Azure App Services
- Microsoft Azure ? Enable Local Cache on Azure App
- Microsoft Azure ? Exploring the Azure Mobile App Home Screen
- Microsoft Azure ? Container Registration for App Deployment
- Microsoft Azure ? Roll Back Versions of Azure Logic Apps
- Microsoft Azure ? Modifying API Connections With Logic Apps
- Microsoft Azure ? Azure Media Service
- Microsoft Azure ? Using Github Action in Azure App Service
- Microsoft Azure ? Creating an HTTP Trigger with Azure Logic Apps
- Microsoft Azure ? Using JSON with Azure Logic Apps
- Microsoft Azure ? Configuring Log Analytics Data Retention
- Microsoft Azure ? Creating JSON Snippet For Policy Initiative Creation using CSV File
- Microsoft Azure ? Working with Files in Azure App Service
- Microsoft Azure ? Using Power Apps in VS Code

Resources

- Microsoft Azure ? Preventing Changes to Resources in Azure App Services
- Microsoft Azure- Writing Queries and Creating Dashboards using Azure Resource Graph
- Microsoft Azure ? Enabling Resource Providers for Azure Subscription
- Microsoft Azure ? Find Orphaned Disks
- Microsoft Azure ? Handling Orphaned NSGs
- Microsoft Azure ? Configuring an Alert Rule with Azure Monitor
- Microsoft Azure ? Count of Azure Resources using Resource Graph Query
- Microsoft Azure ? Create Alert For Conditional Access Policy Changes
- Microsoft Azure ? Create Pie Charts For Orphaned Resources using KQL
- Microsoft Azure ? Patch Management Update Summary Status

- Microsoft Azure ? Check Resource Owner in Azure using KQL
- Microsoft Azure ? Check Enabled and Disabled Diagnostics for Azure Resources
- Microsoft Azure ? Azure Firewall Flow Logs From Select Source IP
- Microsoft Azure ? Resource Tagging and Best Practices
- Microsoft Azure ? Using the Azure Activity Log
- Microsoft Azure ? Scaling Inventory Resources
- Microsoft Azure ? Moving Resources Among Groups

Cognitive Services

- Build Text Translator Webapp using Flask and Cognitive Service
- Microsoft Azure ? Introduction to Metric Advisor
- Microsoft Azure ? Cognitive Service Text Analytics in Web App Containers
- Microsoft Azure ? Opinion Mining with Azure Cognitive Services

Dashboards

- Microsoft Azure ? Customizing Charts on Azure Dashboard
- Microsoft Azure ? Customizing Azure Dashboard Tiles
- Microsoft Azure ? Auto Refreshing Dashboards
- Microsoft Azure ? Pinning to Portal Dashboard

Databases

- Database availability
- Azure SQL Database
- Security in Azure SQL
- Microsoft Azure ? Identifying & Accessing Management for Azure SQL
- Microsoft Azure ? Security Capabilities and Tasks of Azure SQL
- Microsoft Azure ? Configure Azure SQL with Azure CLI
- Microsoft Azure ? Configure Auditing for Azure SQL Database

- Microsoft Azure ? Azure SQL Managed Instance Deployment Options
- Microsoft Azure ? Azure SQL Managed Instance
- Microsoft Azure ? PaaS vs. IaaS in Azure SQL
- Microsoft Azure ? Management Interfaces for Azure SQL
- Microsoft Azure ? Opportunities in the Cloud with Azure SQL
- Microsoft Azure ? Create Azure SQL Database
- Microsoft SQL Server Active Directory Authentication on Linux Machine
- Microsoft Azure ? Introduction to Azure Database For MySQL
- Introduction to Azure Cosmos DB
- Microsoft Azure ? Notebooks in Cosmos DB
- Microsoft Azure ? Using C# in CosmoDB
- Microsoft Azure ? Choosing a Partition Key in Cosmos DB
- Microsoft Azure ? Introduction to Azure Data Factory
- Microsoft Azure ? Migrating From MySQL to Cloud
- Microsoft Azure ? Advanced Threat Protection for Azure SQL Database
- Microsoft Azure ? Using Flexible Server in Azure Database for PostgreSQL
- Microsoft Azure ? Accelerating Cloud Migration & Datacenter Discovery

DevOps

- Introduction to Azure DevOps
- Azure availability for Kubernetes
- Zone Redundancy and SLA of Azure SQL
- Create a Project in Azure DevOps
- Create a Project in Azure using the Agile Process
- Microsoft Azure ? Starting & Stopping an Azure Kubernetes Service Cluster
- Microsoft Azure ? Introduction to Azure Kung Fu Toolkit
- Microsoft Azure ? Introduction to Kubernetes Diagnostics

- How to Create Backlog and Sprint in Agile Process?
- Microsoft Azure ? Running an App inside a Docker Container Image
- How to Push a Container Image to a Docker Repository?
- Microsoft Azure ? Manage and Export Azure Policies with GitHub Integration
- Microsoft Azure Arc Enabled Kubernetes
- Microsoft Azure ? Introduction to Azure Arc
- Microsoft Azure ? Deploying Web Applications Using GitHub Actions
- Microsoft Azure ? Messaging with Azure Web PubSub
- Microsoft Azure ? Testing in Production Sites Feature using PowerShell
- Microsoft Azure ? Build and Deploy App with Azure SDK For Java

Machine Learning and IoT

- Microsoft Azure ? Introduction to Azure Sentinel
- Machine Learning with Microsoft Azure ML Studio Without Code
- What is the Best Azure IoT Service for You?
- Microsoft Azure ? Key IoT Products
- Microsoft Azure ? Using Azure Form Recognizer

Miscellaneous

- Microsoft Azure ? Productivity Tips For Azure portal
- Microsoft Azure ? Get Azure Policies Definitions and Policy Initiatives Definitions
- Microsoft Azure ? Create a Custom Policy Initiative in Azure
- Microsoft Mesh
- Microsoft Azure ? Working with Azure Health Bot
- Microsoft Azure ? Introduction to Azure Digital Twins
- Microsoft Azure ? Working with Themes in Azure
- Microsoft Azure ? Understanding Azure Keyboard Shortcuts
- Microsoft Azure ? Using Table Parameter in Azure CLI

- Microsoft Azure ? Add Azure Cloud Shell to VS Code
- Microsoft Azure ? Accessing Cloud Shell from Microsoft Docs
- Microsoft Azure ? Azure App Servers Routing using PowerShell
- Microsoft Azure ? Introduction to SMART
- Microsoft Azure ? Lift & Shift Approach for Infrastructure Migration
- Microsoft Azure ? Configure Diagnostic Settings For Azure Subscription
- Microsoft Azure ? Checking your Access on Azure Subscription
- Microsoft Azure ? Enabling Logs for Troubleshooting the Azure Firewall Rules
- Microsoft Azure ? Rebooting an Application Gateway
- Microsoft Azure ? PAL Tagging in Azure
- Microsoft Azure ? Using Cost Management in Azure Portal
- Microsoft Azure ? Finding the Right Load Balancing Service
- Microsoft Azure ? View Delegated Roles in Azure Lighthouse
- Microsoft Azure ? Using the Global Search
- Microsoft Azure ? Quick Copy & Paste on Portal
- Microsoft Azure ? Alerts as Notifications on Phone
- Microsoft Azure ? Developing ARM Templates using Azure Bicep

Features of Microsoft Azure

Microsoft Azure is a comprehensive cloud computing platform developed by Microsoft. Here are some of its key features:

Features of Microsoft Azure

- Cloud Services: Azure provides services like Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). It supports many programming languages, frameworks, and tools, including third-party systems.
- Computational Services: Azure's Infrastructure as a Service (IaaS) feature allows you to launch general-purpose virtual machines in Microsoft Windows and Linux platforms.

- Data Resilience: Azure Cloud services offer more distribution points and data centers, enhancing data resilience.
- Data Security: Microsoft Azure has many levels of built-in security to meet customer needs.
- Integrated Networking: Networking is an important part of Microsoft Azure feature in Cloud architecture.
- Enhanced Infrastructure Management: Azure provides tools for managing infrastructure, improving efficiency.
- Azure IoT Feature: Azure offers Internet of Things (IoT) services, enabling devices to connect and interact.
- Deployment Models: Azure offers two deployment models for cloud resources ? Classic Deployment Model and Azure Resource Manager.
- Wide Variety of Services: Azure delivers more than 600 services¹.
- Flexibility and Speed: Azure is known for its enhanced flexibility, integrated delivery pipeline, security, disaster recovery, and speed of service.

Conclusion

With the numerous functionalities and features outlined in this lesson, the Microsoft Azure cloud platform improves business efficiency. Additionally, it is affordable and easy for beginners to understand. Also, this tutorial will help learners in preparation for Azure certifications. Here we will see how the storage, compute, and network services will work along with the management and deployment of data in the cloud.

What is Microsoft Azure?

Microsoft Azure is a cloud computing platform and infrastructure created by Microsoft. It provides a wide range of services including computing, analytics, storage, and networking. Azure allows users to build, deploy, and manage applications and services through Microsoft-managed data centers.

Why use Microsoft Azure?

Microsoft Azure offers several benefits, including scalability, high availability, and flexibility. It allows businesses to easily scale their infrastructure based on demand, provides redundancy and failover options for increased reliability, and supports a wide range of programming languages, frameworks, and operating systems.

How to get started with Microsoft Azure?

To get started with Microsoft Azure, you can sign up for an Azure account and access the free services and resources available for beginners. Microsoft also provides comprehensive documentation, tutorials, and training resources to help users understand and leverage the capabilities of Azure.

What are the main features of Microsoft Azure?

Microsoft Azure offers a variety of features, including virtual machines, app services, storage options, databases, networking solutions, and analytics tools. It also provides AI and machine learning services, IoT capabilities, and comprehensive security and compliance features.

Can I migrate my existing applications to Microsoft Azure?

Yes, Microsoft Azure provides tools and services to help businesses migrate their existing applications, workloads, and infrastructure to the Azure cloud. The platform supports various migration strategies and offers compatibility with a wide range of technologies and frameworks.

How does Microsoft Azure ensure security and compliance?

Microsoft Azure implements robust security measures and supports a wide range of compliance certifications and standards. It offers built-in security controls, threat intelligence, and compliance tools to help users protect their data and meet industry-specific regulatory requirements.

How does Microsoft Azure ensure security and compliance?

How does Microsoft Azure ensure security and compliance?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production

course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

How does Microsoft Azure ensure security and compliance?

- DevOps
- Microsoft Azure
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest

- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

REST API Introduction

-
-
-

REST API Introduction

REpresentationalStateTransfer (REST) is an architectural style that defines a set of constraints to be used for creating web services. REST API is a way of accessing web services in a simple and flexible way without having any processing.

REST API Introduction

REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST uses less bandwidth, simple and flexible making it more suitable for internet usage. It's used to fetch or give some information from a web service. All communication done via REST API uses only HTTP request.

REST API Introduction

Working: A request is sent from client to server in the form of a web URL as HTTP GET or POST or PUT or DELETE request. After that, a response comes back from the server in the form of a resource which can be anything like HTML, XML, Image, or JSON. But now JSON is the most popular format being used in Web Services.

REST API Introduction

REST API Introduction

In HTTP there are five methods that are commonly used in a REST-based Architecture i.e., POST, GET, PUT, PATCH, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations respectively. There are other methods which are less frequently used like OPTIONS and HEAD.

REST API Introduction

- GET: The HTTP GET method is used to read (or retrieve) a representation of a resource. In the safe path, GET returns a representation in XML or JSON and an HTTP response code of 200 (OK). In an error case, it most often returns a 404 (NOT FOUND) or 400 (BAD REQUEST).
- POST: The POST verb is most often utilized to create new resources. In particular, it's used to create subordinate resources. That is, subordinate to some other (e.g. parent) resource. On successful creation, return HTTP status 201, returning a Location header with a link to the newly-created resource with the 201 HTTP status.

REST API Introduction

NOTE: POST is neither safe nor idempotent.

REST API Introduction

- PUT: It is used for updating the capabilities. However, PUT can also be used to create a resource in the case where the resource ID is chosen by the client instead of by the server. In other words, if the PUT is to a URI that contains the value of a non-existent resource ID. On successful update, return 200 (or 204 if not returning any content in the body) from a PUT. If using PUT for create, return HTTP status 201 on successful creation. PUT is not safe operation but it's idempotent.
- PATCH: It is used to modify capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource. This resembles PUT, but the body contains a set of instructions describing how a resource currently residing on the server should be modified to produce a new version. This means that the PATCH body should not just be a modified part of the resource, but in some kind of patch language like JSON Patch or XML Patch. PATCH is neither safe nor idempotent.
- DELETE: It is used to delete a resource identified by a URI. On successful deletion, return HTTP status 200 (OK) along with a response body.

REST API Introduction

Idempotence: An idempotent HTTP method is a HTTP method that can be called many times without

different outcomes. It would not matter if the method is called only once, or ten times over. The result should be the same. Again, this only applies to the result, not the resource itself.

REST API Introduction

Example:

Request and Response

Now we will see how request and response work for different HTTP methods. Let's assume we have an API (<https://www.geeksforgeeks.org/api/students>) for all students data of gfg.

Request and Response

- GET: Request for all Students.

Request and Response

Request

GET:/api/students

Request and Response

- POST: Request for Posting/Creating/Inserting Data

Request and Response

Request

POST:/api/students{?name?:?Raj?}

Request and Response

Request

Request and Response

POST:/api/students

Request and Response

{?name?:?Raj?}

Request and Response

- PUT or PATCH:Request for Updating Data at id=1

Request and Response

Request

PUT or PATCH:/api/students/1{?name?:?Raj?}

Request and Response

Request

Request and Response

PUT or PATCH:/api/students/1

Request and Response

{?name?:?Raj?}

Request and Response

- DELETE:Request for Deleting Data of id=1

Request and Response

Request

DELETE:/api/students/1

Request and Response

RESTful web services are very popular because they are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

Request and Response

Feeling lost in the vast world of Backend Development? It's time for a change! Join ourJava

Backend Development - Live Course and embark on an exciting journey to master backend development efficiently and on schedule. What We Offer:

Request and Response

- Comprehensive Course
- Expert Guidance for Efficient Learning
- Hands-on Experience with Real-world Projects
- Proven Track Record with 100,000+ Successful Geeks

Request and Response

- Web Technologies
- RESTful
- Web-API

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore

- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap

- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is Low Level Design or LLD? ? Learn System Design

-
-
-

What is Low Level Design or LLD? ? Learn System Design

Low-Level Design (LLD) is the detailed design process in the software development process that focuses on implementing individual components described in the High-Level Design. It provides a blueprint for how each component in the system will function and process and it also includes UML Diagrams, data structures, and algorithms. The input to LLD is HLD i.e. LLD translates the HLD into smaller and more specific details.

What is Low Level Design or LLD? ? Learn System Design

What is Low Level Design or LLD?

What is Low Level Design or LLD? ? Learn System Design

Table of Content

What is Low Level Design or LLD? ? Learn System Design

- What is Low-Level Design(LLD)?
- How is LLD different from HLD?
- Roadmap to form LLD from HLD?
- Benefits of Low-Level Design(LLD)
- Best Practices for Low-Level Design(LLD)
- Roadmap to learn LLD

What is Low-Level Design(LLD)?

LLD, or Low-Level Design, is a phase in the software development process where detailed system

components and their interactions are specified, along with their implementation. It involves converting the high-level design into a more detailed blueprint, addressing specific algorithms, data structures, and interfaces. LLD serves as a guide for developers during coding, which ensures the accurate and efficient implementation of the system's functionality.

What is Low-Level Design(LLD)?

Low-level designing is also known as object-level designing or micro-level or detailed designing.

How is LLD different from HLD?

High Level Design or HLD is a general system design where we do tradeoffs between different frameworks, components, and different databases and we choose the best considering what the business needs and how the system should work, both in terms of functional and non functional aspects.

How is LLD different from HLD?

Whereas LLD(Low Level Design), translates the HLD into smaller and more specific details, It includes class diagrams, methods, data structures, and algorithms, focusing on how each part will be implemented. It is Primarily used by developers and technical teams.

How is LLD different from HLD?

Further Read: Differences between High Level Design(HLD) and Low Level Design(LLD)

Roadmap to form LLD from HLD?

As we know, input for framing low-level design (LLD) is (High-Level Design) HLD. For this conversion, we generally use Unified Modelling Language (UML) diagrams. Adding to these diagrams we use OOPS principles and SOLID principles and design patterns while designing. Hence, using these paradigms we can convert any HLD to LLD so as to get implemented.

Roadmap to form LLD from HLD?

In order to understand how to design any low-level diagram let us understand via the steps:

Roadmap to form LLD from HLD?

Step 1. Object-oriented Principles

The user requirement is processed by using concepts of OOPS programming. Hence it is recommended to have a strong grip on OOPS concepts prior to moving ahead in designing any low-level system. Programmers should be absolutely clear about the OOPS concepts to depth right to classes, and objects because OOPS is the foundation on which low-leveling on any system is based.

Step 2. Process of analyzing and design

It is a analyzing phase which is our 1st step where we are forming real-world problems into object-world problems using OOPS concepts and SOLID principles.

Step 3. Design Patterns

Now the implementation of our above object oriented problem is carried out with the help of design patterns. Design patterns are reusable solutions to common problems encountered in software design. They provide a structured approach to design by capturing best practices and proven solutions, making it easier to develop scalable, maintainable, and efficient software.

Step 4. UML Diagram

UML (Unified Modeling Language) diagrams play an important role in converting HLD to LLD. They provide a proper and clear visual representation of the components and their relationships, which helps developers significantly

5. SOLID Principles

These are sets of 5 principles(rules) that are strictly followed as per requirements of the system or requirements for optimal designing. In order to write scalable, flexible, maintainable, and reusable code:

5. SOLID Principles

- Single-responsibility principle (SRP)
- Open-closed principle (OCP)
- Liskov's Substitution Principle(LSP)
- Interface Segregation Principle (ISP)
- Dependency Inversion Principle (DIP)

5. SOLID Principles

It's important to keep in mind that SOLID principles are just guidelines and not strict rules to be followed. The key is to strike a balance between following these principles and considering the specific needs and constraints of your business requirement.

Benefits of Low-Level Design(LLD)

Let us discuss some of the benefits of Low-Level Design:

Benefits of Low-Level Design(LLD)

- LLD provides a detailed plan for how each part of the software will work
- It provides a well-structured design, which makes simpler to update or fix parts of the system without affecting the entire software.
- LLD helps team members communicate more effectively because everyone has a clear understanding of how components are working.
- Following the design principles in LLD leads to more cleaner, organized code, making it less prone to errors.
- A proper LLD speeds up the coding process because developers can follow the detailed plan made earlier.

Best Practices for Low-Level Design(LLD)

Below are some of the best practices to form a good Low-Level Design:

Best Practices for Low-Level Design(LLD)

- Always try to break down the system into small, independent components that contain specific functionalities.
- Clearly define the interfaces for each component, which should include methods, inputs, outputs. This helps in maintaining proper communication between components
- Include OOP Principles to promote code reusability, flexibility and maintainability.
- Follow Solid Principles which will lead to more robust and maintainable design.
- plan in advance for error handling and validation of the system by including validation checks in the design

1. Object-Oriented and Design Principles

- Object-Oriented Programming(OOP) Concepts
- Object-Oriented Analysis and Design
- SOLID Principles

2. Design Patterns

- Creational Design Patterns
- Behavioral Design Patterns
- Structural Design Patterns
- When to choose which design pattern?

3. UML and Modeling in LLD

- Class Diagrams
- Sequence Diagrams
- Activity Diagrams
- State Diagrams
- Use Case Diagrams
- Object Diagrams

4. LLD Best Practices

- Data Structures and Algorithms for System Design
- Essential Security Measures in System Design
- Introduction to Modularity and Interfaces
- Low-Level Design Document
- Difference between LLD and HLD
- Performance vs Scalability
- Speed vs. Quality
- Coding Standards Best Practices

5. LLD Case Studies

- Elevator System Low-Level Design
- Tic-Tac-Toe Low-Level Design
- Amazon Low Level Design
- Vending Machine Low Level Design

6. Interview Preparation

- How to Prepare for Low-Level Design Interviews?
- Top Low-Level Design(LLD) Interview Questions 2024
- Best Books for Learning Low-Level Design(LLD) [2024]

Conclusion

In conclusion, Low-Level Design (LLD) is an important phase in the software development lifecycle that translates high-level architectural concepts into detailed specifications for individual components. By focusing on modularity, clear interface definitions, and following the design principles, LLD ensures that software systems are robust, maintainable, and scalable.

Conclusion

Become a System Design expert and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Conclusion

Step into the Three 90 Challenge ? 90 days to push limits, break barriers, and become the best version of yourself! Don't miss your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Conclusion

- System Design

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design

- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python

- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping

- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams

- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL

- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials

- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]

- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is High Level Design? ? Learn System Design

-

-

-

What is High Level Design? ? Learn System Design

In Developing scalable applications, proper planning, and organization play a significant role. High-level design plays an important role in this process by serving as the blueprint of the system's architecture. It provides a comprehensive view of how components interact and function together which will further help for detailed implementation.

What is High Level Design? ? Learn System Design

Understanding high-level design is very crucial for developers, architects, and project managers because it allows them to make sure that all stakeholders are aligned with the project objectives. High-level design is also known as macro-level design.

What is High Level Design? ? Learn System Design

What is High-Level Design

What is High Level Design? ? Learn System Design

Table of Content

What is High Level Design? ? Learn System Design

- What is High-Level Design?
- Components of High-Level Design
- What is High-Level Design Document?
- Purpose and Characteristics of High-Level Design
- How HLD is different from LLD
- How To Design Scalable High-level Design (HLD) Systems
- Roadmap to learn HLD

What is High-Level Design?

High-Level Design is a initial step in development of applications where overall structure of a system is planned. High-Level design focuses mainly on how different components of the system work together without getting to know about internal coding and implementation. This helps everyone involving in the project to understand the goals and ensures good communication during development.

What is High-Level Design?

Components of High-Level Design

Components of High-Level Design

Understanding the components of high-level design is very important for creating effective systems that meets user needs and technical requirements. Below are the main components of high-level design:

Components of High-Level Design

- **System Architecture:** System architecture is an overview of the entire system which represents the structure and the relationships between various components. It helps to visually represent how different parts interact and function.
- **Modules and Components:** High-Level design breaks down the systems into modules or components each with specific roles and responsibilities, and has a distinct function that contributes to entire system helping in developing an efficient system.
- **Data Flow Diagrams (DFDs):** Data Flow Diagrams demonstrates the data movement within the system. They help to understand how information is processed and handled.
- **Interface Design:** This component focuses on how different modules communicate with one another. It details the application programming interfaces (APIs) and user interfaces necessary for seamless interaction between components.
- **Technology Stack:** The technology stack are various technologies and tools that will be used in the development of the system. This includes programming languages, frameworks, databases.

- Deployment Architecture: It includes how the system will be hosted and accessed. It includes server configurations, cloud infrastructure, and network considerations.

What is High-Level Design Document?

HLD document consists of data flows, flowcharts, and data structures to help developers in understanding and implement how the current system is being designed intentionally to function. This document is responsible for explaining the connections between system components and operations which depict the logic. The architecture design needed (for the system's functionality and flow) for each and every module of the system as per the functional requirements.

Purpose and Characteristics of High-Level Design

The purpose of this High-Level Design (HLD) is to add the necessary detailed description to represent a suitable model. This is designed to help with operational requirements and will help to understand how the modules interact. Basically, HLD is a technical representation of functional requirements and the flow of information across components. Characteristics of High-Level Design include:

Purpose and Characteristics of High-Level Design

- A diagram representing each design aspect is included in the HLD (which is based on business requirements and anticipated results).
- Description of hardware, software interfaces, and also user interfaces.
- The workflow of the user's typical process is detailed, along with performance specifications.

How HLD is different from LLD

High Level Design or HLD is a general system design where we do tradeoffs between different frameworks, components, and different databases and we choose the best considering what the business needs and how the system should work, both in terms of functional and non functional aspects.

How HLD is different from LLD

Whereas LLD(Low Level Design), translates the HLD into smaller and more specific details, It includes class diagrams, methods, data structures, and algorithms, focusing on how each part will be implemented.

How HLD is different from LLD

Further Read:Differences between High Level Design(HLD) and Low Level Design(LLD)

How HLD is different from LLD

After having an adequate understanding of HLD and how it is different from LLD. Let us now discuss the HLD roadmap, which is shown below as an infographic:

How To Design Scalable High-level Design (HLD) Systems

Now in order to design any high-level system, certain terms are listed in a sequential manner so as to connect the dots in order to understand, Roadmap is very important for working professionals because these help us to get a complete binding understanding of how services in real-time are getting scaled at a high level.

How To Design Scalable High-level Design (HLD) Systems

How To Design Scalable High-level Design (HLD) Systems

1.Capacity Estimation

Capacity estimation in system design involves predicting the resources (such as processing power, memory, and bandwidth) required to meet the expected workload. It ensures that a system can handle current and future demands efficiently, helping in the proper allocation of resources and preventing performance bottlenecks.

1.Capacity Estimation

For example:

1.Capacity Estimation

Twitter which is recently in newsfeeds these days, here at high-level designing we need to make sure tweets of popular influencers are reaching out to millions of people so how we need to scale our system so service should not be hampered.

1.Capacity Estimation

We have also shown below the characteristics behavior of a server which is measured across throughput and latency within the system.

1.Capacity Estimation

Comparison of Throughput and Latency with Concurrency : Behavior Of A Server

2. HTTP and HTTPS and their methods

HTTP(HyperText Transfer Protocol), is used to transfer the data over the web, which enables the communication between clients and server, that helps user to requests resources like HTML pages, messages, videos or images while HTTPS(HyperText Transfer Protocol Secure) is an extension of HTTP which adds a extra layer of security through SSL/TLS encryption. Methods of HTTP include GET, PUT and POST.

2. HTTP and HTTPS and their methods

Tip:These are also important as per interview perceptive as constraint bounds in interviews in API over when to use what is very important.

2. HTTP and HTTPS and their methods

Tip:There are 2 ways to send data from server to client:

2. HTTP and HTTPS and their methods

- Using Websockets
- Using Polling

3.Web Sockets

There are two way to send data from server to clients which are websockets and polling. In a client-server architecture, we are sending a request to server and server sends it back and so in this way communication takes place. But in designing the system biggest problem we face is whether we can implement the client-server model because we can not wait until the server responds.

3.Web Sockets

- Here Web Sockets play a crucial role that solving problems and enabling us to have real-time communication.
- WebSocket is a full-duplex protocol as it allows the application to send and receive data at the same time.
- They are more or less used in every system to be designed because we can not let our system wait for a response.
- They are very useful in Real-time web applications, gaming applications, chat applications.

3.Web Sockets

WebSocket Connection

4.Polling

Another way of sending and receiving data from a server just likely as we do we doing above in the case of web sockets Polling. There are mainly 2 types of polling:

4.Polling

- In short polling, we sent the request. In short polling, the client sends a request to the server and the server immediately sends data at fixed intervals whether the data is updated or not.
- In long polling, the server waits for the client's request to respond and responds only and only if some new data is available or if some data is being updated.

4.Polling

Short polling vs long polling

4.Polling

Tip:Long polling is preferredover short polling because lesser number of requests are sent in a system.

5. Server-Sent Events(SSE)

It is purposely built as one-way communication from servers to clients in specific design systems. SSE is a technology that helps to push real-time updates from the server to the clients over HTTP connection. SSE enables server to send data automatically as it becomes available.Example of SSE include real-time streaming.

5. Server-Sent Events(SSE)

Server-Sent Events(SSE)

6.Rate limiting

A rate limiter restricts the number of events that can be done in a timeframe by restricting the number of requests a sender can send in a given period of time. Here once the threshold limit is reached now further it blocks the incoming requests as can be seen below media as follows:

6.Rate limiting

Rate limiting

7.Resiliency

No matter how great a system we design there is always a chance of faults and failure which could be because of hardware issues or software issues(such as running low on memory) or there can be some human error. In such cases we need to provide resiliency through replication, redundancy and availability.

7. Resiliency

Note: System design interviews start with open-ended designing a specific system which later is bounded with constraints at multiple levels. In order to deal with these bounds in layman language is known as resiliency via implementing common tradeoff in the system. Hence providing resiliency is very crucial in designing complex system designs and also in clearing interviews.

8. Paging

Paging in High-Level Design (HLD) refers to the method of dividing large datasets or content into smaller, manageable pages. This approach improves user experience by allowing users to load data incrementally, reducing initial load times and enhancing performance.

8. Paging

Paging

9. Logging

A log file records details of events occurring in a software application. The details may consist of microservices, transactions, service actions, or anything helpful to debug the flow of an event in the system. Logging is crucial to monitor the application's flow. This can also be useful for tracking the health and performance of a distributed system, as well as for debugging issues that may arise. There are several approaches to implementing distributed logging, including:

9. Logging

- Using a centralized logging service
- Using a distributed logging system
- Using a cloud-based logging service

1. Basics of HLD

- Difference between HLD and LLD
- What is the goal of High-Level Design (HLD)?

2. Core Components of HLD

- Load Balancer
- Content Delivery Network(CDN)
- Caching in System Design
- What is API Gateway
- Message Queues
- Consistent Hashing
- Communication Protocols
- Network Protocols and Proxies

3. Design Principles for HLD

- Scalability and How to achieve it
- Availability in System Design
- Consistency in System Design
- Reliability in System Design
- Fault Tolerance in System Design
- CAP Theorem
- Performance Optimization Techniques for System Design
- How to Draw High Level Design Diagram?

4. Types of HLD Diagrams

- Component Based Diagram
- Sequence Diagrams
- What is DFD(Data Flow Diagram)?
- Deployment Diagram

5. Architectures and Patterns

- Monolithic Architecture

- Microservices Architecture
- Event-Driven Architecture
- Client-Server Architecture
- Peer-to-Peer (P2P) Architecture
- Serverless Architecture
- Circuit Breaker Pattern
- Event Sourcing Pattern
- CQRS Design Pattern

6. Interview Preparation for HLD

- How to Crack System Design Interview Round?
- Top High-Level Design(HLD) Interview Questions 2024

6. Interview Preparation for HLD

6. Interview Preparation for HLD

Become a System Design expert and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

6. Interview Preparation for HLD

Step into the Three 90 Challenge ? 90 days to push limits, break barriers, and become the best version of yourself! Don't miss your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

6. Interview Preparation for HLD

- System Design

Similar Reads

Similar Reads

- [Company](#)
- [About Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)

- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering

- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics

- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation

- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]

- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course

- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Computer Vision

- Computer Vision - Introduction
- A Quick Overview to Computer Vision
- Applications of Computer Vision
- Fundamentals of Image Formation
- Satellite Image Processing
- Image Formats

Image Processing & Transformation

- Digital Image Processing Basics
- Difference Between RGB, CMYK, HSV, and YIQ Color Models
- Image Enhancement Techniques using OpenCV - Python

- Image Transformations using OpenCV in Python
- How to find the Fourier Transform of an image using OpenCV Python?
- Python | Intensity Transformation Operations on Images
- Histogram Equalization in Digital Image Processing
- Python - Color Inversion using Pillow
- Image Sharpening Using Laplacian Filter and High Boost Filtering in MATLAB
- Wand sharpen() function - Python
- Python OpenCV - Smoothing and Blurring
- Python PIL | GaussianBlur() method
- Apply a Gauss filter to an image with Python
- Spatial Filtering and its Types
- Python PIL | MedianFilter() and ModeFilter() method
- Python | Bilateral Filtering
- Python OpenCV - Morphological Operations
- Erosion and Dilation of images using OpenCV in python
- Introduction to Resampling methods
- Python | Image Registration using OpenCV

Feature Extraction and Description

- Feature Extraction Techniques - NLP
- SIFT Interest Point Detector Using Python - OpenCV
- Feature Matching using Brute Force in OpenCV
- Feature detection and matching with OpenCV-Python
- Feature matching using ORB algorithm in Python-OpenCV
- Mahotas - Speeded-Up Robust Features
- Create Local Binary Pattern of an image using OpenCV-Python

Deep Learning for Computer Vision

- Image Classification using CNN
- What is Transfer Learning?
- Top 5 PreTrained Models in Natural Language Processing (NLP)
- ML | Introduction to Strided Convolutions
- Dilated Convolution
- Continuous Kernel Convolution
- CNN | Introduction to Pooling Layer
- CNN | Introduction to Padding
- What is the difference between 'SAME' and 'VALID' padding in tf.nn.max_pool of tensorflow?
- Convolutional Neural Network (CNN) Architectures
- Deep Transfer Learning - Introduction
- Introduction to Residual Networks
- Residual Networks (ResNet) - Deep Learning
- ML | Inception Network V1
- Understanding GoogLeNet Model - CNN Architecture
- Image Recognition with Mobilenet
- VGG-16 | CNN model
- Autoencoders -Machine Learning
- How Autoencoders works ?
- Difference Between Encoder and Decoder
- Implementing an Autoencoder in PyTorch
- Generative Adversarial Network (GAN)
- Deep Convolutional GAN with Keras
- StyleGAN - Style Generative Adversarial Networks

Object Detection and Recognition

- Detect an object with OpenCV-Python

- Python | Haar Cascades for Object Detection
- R-CNN - Region-Based Convolutional Neural Networks
- YOLO v2 - Object Detection
- Face recognition using Artificial Intelligence
- Deep Face Recognition
- ML | Face Recognition Using Eigenfaces (PCA Algorithm)
- Emojify using Face Recognition with Machine Learning
- Object Detection with Detection Transformer (DETR) by Facebook

Image Segmentation

- Image Segmentation Using TensorFlow
- Thresholding-Based Image Segmentation
- Region and Edge Based Segmentation
- Image Segmentation with Watershed Algorithm - OpenCV Python
- Mask R-CNN | ML

3D Reconstruction

- Python OpenCV - Depth map from Stereo Images
- Top 7 Modern-Day Applications of Augmented Reality (AR)
- Virtual Reality, Augmented Reality, and Mixed Reality
- Camera Calibration with Python - OpenCV
- Python OpenCV - Pose Estimation

Computer Vision Tutorial

-
-
-

Computer Vision Tutorial

Computer vision is a field of study within artificial intelligence (AI) that focuses on enabling computers to Intercept and extract information from images and videos, in a manner similar to human vision. It involves developing algorithms and techniques to extract meaningful information from visual inputs and make sense of the visual world.

Computer Vision Tutorial

This Computer Vision tutorial is designed for both beginners and experienced professionals, covering both basic and advanced concepts of computer vision, including Digital Photography, Satellite Image Processing, Pixel Transformation, Color Correction, Padding, Filtering, Object Detection and Recognition, and Image Segmentation.

Computer Vision Examples

- Facial recognition: Identifying individuals through visual analysis.
- Self-driving cars: Using computer vision to navigate and avoid obstacles.
- Robotic automation: Enabling robots to perform tasks and make decisions based on visual input.
- Medical anomaly detection: Detecting abnormalities in medical images for improved diagnosis.
- Sports performance analysis: Tracking athlete movements to analyze and enhance performance.
- Manufacturing fault detection: Identifying defects in products during the manufacturing process.
- Agricultural monitoring: Monitoring crop growth, livestock health, and weather conditions through visual data.

Computer Vision Examples

Prerequisite: Before Starting Computer Vision It's Recommended that you should have a foundational knowledge of Machine Learning, Deep learning and an OpenCV. you can refer to our tutorial page on prerequisites technologies.

Mathematical prerequisites for Computer Vision

1. Linear Algebra

Mathematical prerequisites for Computer Vision

- Vectors
- Matrices and Tensors
- Eigenvalues and Eigenvectors
- Singular Value Decomposition

Mathematical prerequisites for Computer Vision

2.Probability and Statistics

Mathematical prerequisites for Computer Vision

- Probability Distributions
- Bayesian Inference and Bayes' Theorem
- Markov Chains
- Kalman Filters

Mathematical prerequisites for Computer Vision

3.Signal Processing

Mathematical prerequisites for Computer Vision

- Image Filtering and Convolution
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Principal Component Analysis (PCA)

Image Processing

Image processing refers to a set of techniques for manipulating and analyzing digital images. The techniques include:

Image Processing

1.Image Transformationis process of modifying or changing an images.

Image Processing

- Geometric Transformations
- Fourier Transform
- Intensity Transformation

Image Processing

2. Image Enhancement improve the visual quality or clarity of image to highlight important features or details to minimize noise or distortions.

Image Processing

- Histogram Equalization
- Contrast Enhancement
- Image Sharpening
- Color Correction

Image Processing

3. Noise Reduction Techniques removes unwanted noise from images while preserving important features like edges and texture.

Image Processing

- Gaussian Smoothing
- Median Filtering
- Bilateral Filtering
- Wavelet Denoising

Image Processing

4. Morphological Operations process images based on their structure and shape. Common morphological operations include:

Image Processing

- Erosion and Dilation
- Opening
- Closing
- Morphological Gradient

Feature Extraction

1.Edge Detection Techniquesidentify significant changes in the intensity or color, that corresponds to the boundaries of objects with an image.

Feature Extraction

- Canny Edge Detector
- Sobel Operator
- Prewitt Operator
- Laplacian of Gaussian (LoG)

Feature Extraction

2.Corner and Interest Point Detectionidentify points in an image that are distinctive and can be detected across different views, transformations or scales.

Feature Extraction

- Harris Corner Detection
- Shi-Tomasi Corner Detector

Feature Extraction

3.Feature Descriptorsgenerates a compact representation of local image region around keypoints making it easier to correspond features across different images.

Feature Extraction

- SIFT (Scale-Invariant Feature Transform)
- SURF (Speeded-Up Robust Features)

- ORB (Oriented FAST and Rotated BRIEF)
- HOG (Histogram of Oriented Gradients)

Deep Learning for Computer Vision

Deep learning has revolutionized the field of computer vision by enabling machines to understand and interpret visual data in ways that were previously unimaginable.

1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks are designed to learn spatial hierarchies of features from images. Key components include:

1. Convolutional Neural Networks (CNNs)

- Convolutional Layers
- Pooling Layers
- Fully Connected Layers

2. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) consist of two networks (generator and discriminator) that work against each other to create realistic images. There are various types of GANs, each designed for specific tasks and improvements:

2. Generative Adversarial Networks (GANs)

- Deep Convolutional GAN (DCGAN)
- Conditional GAN (cGAN)
- Cycle-Consistent GAN (CycleGAN)
- Super-Resolution GAN (SRGAN)
- Wasserstein GAN (WGAN)
- StyleGAN

3. Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are probabilistic version of autoencoders, which forces the model to learn a distribution over the latent space rather than a fixed point. Other autoencoders used in computer vision are:

3. Variational Autoencoders (VAEs)

- Vanilla Autoencoders
- Denoising Autoencoders (DAE)
- Convolutional Autoencoder (CAE)

4. Vision Transformers (ViT)

Vision Transformers (ViT) are inspired by transformers models to treat images and sequence of patches and process them using self-attention mechanisms. Common vision transformers include:

4. Vision Transformers (ViT)

- DeiT (Data-efficient Image Transformer)
- Swin Transformer
- CvT (Convolutional Vision Transformer)
- T2T-ViT (Tokens-to-Token Vision Transformer)

5. Vision Language Models

Vision language models integrate visual and textual information to perform image processing and natural language understanding.

5. Vision Language Models

- CLIP (Contrastive Language-Image Pre-training)
- ALIGN (A Large-scale Image and Noisy-text)
- BLIP (Bootstrapping Language-Image Pre-training)

Computer Vision Tasks

1. Image Classification assigns a label or category to an entire image based on its content.

Computer Vision Tasks

- Multiclass classificationclassifies an image into multiple predefined classes.
- Multilabel classificationinvolves assigning multiple labels to a single image.
- Zero-shot classificationclassifies images into categories that model has never seen during training.

Computer Vision Tasks

You can perform image classification using following methods.

Computer Vision Tasks

- Image Classification using Support Vector Machine (SVM)
- Image Classification using RandomForest
- Image Classification using CNN
- Image Classification using TensorFlow
- Image Classification using PyTorch Lightning
- Image Classification using InceptionResNetV2

Computer Vision Tasks

To learn about the datasets for image classification, you can go through the article onDataset for Image Classification.

Computer Vision Tasks

2.Object Detectioninvolves identifying and locating objects within an image by drawing bounding boxes around them. Object detection include following concepts:

Computer Vision Tasks

- Bounding Box Regression
- Intersection over Union (IoU)
- Region Proposal Networks (RPN)
- Non-Maximum Suppression (NMS)

Type of Object Detection Approaches

1. Single-Stage Object Detection

Type of Object Detection Approaches

- YOLO (You Only Look Once)
- SSD (Single Shot Multibox Detector)

Type of Object Detection Approaches

2. Two-Stage Object Detection

Type of Object Detection Approaches

- Region-Based Convolutional Neural Networks (R-CNNs)
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN

Type of Object Detection Approaches

You can perform object detection using the following methods:

Type of Object Detection Approaches

- Object Detection using TensorFlow
- Object Detection using PyTorch

Type of Object Detection Approaches

3. Image Segmentation involves partitioning an image into distinct regions or segments to identify objects or boundaries at a pixel level. Types of image segmentation are:

Type of Object Detection Approaches

- Semantic Segmentation
- Instance Segmentation

- Panoptic Segmentation

Type of Object Detection Approaches

You can perform image segmentation using the following methods:

Type of Object Detection Approaches

- Image Segmentation using K Means Clustering
- Image Segmentation using UNet
- Image Segmentation using UNet++
- Image Segmentation using TensorFlow
- Image Segmentation with Mask R-CNN

Type of Object Detection Approaches

To learn more related to this, you can refer to: [Computer Vision Tasks](#)

Type of Object Detection Approaches

How does Computer Vision Work?

Computer Vision Works similarly to our brain and eye work, To get any Information first our eye capture that image and then sends that signal to our brain. Then After, our brain processes that signal data and converted it into meaningful full information about the object then It recognizes/categorises that object based on its properties.

How does Computer Vision Work?

In a similar fashion to Computer Vision Work, In CV we have a camera to capture the Objects and Then it processes that Visual data by some pattern recognition algorithms and based on that property that object is identified. But, Before giving unknown data to the machine/Algorithm, we trained that machine on a vast amount of Visual labelled data. This labelled data enables the machine to analyze different patterns in all the data points and can relate to those labels.

How does Computer Vision Work?

Example: Suppose we provide audio data of thousands of bird songs. In that case, the computer learns from this data, analyzes each sound, pitch, duration of each note, rhythm, etc., and hence identifies patterns similar to bird songs and generates a model. As a result, this audio recognition model can now accurately detect whether the sound contains a bird song or not for each input sound.

Evolution of Computer Vision

Time Period | Evolution of Computer Vision

2010-2015 | Development of deep learning algorithms for. recognition image.Introduction of convolutional neural networks (CNNs) for image classification.Use of computer vision in autonomous vehicles for object detection and navigation.

2015-2020 | Advancements in real-time object detection with systems like YOLO (You Only Look Once).in facial recognition technology, used in various applications like unlocking smartphones and surveillance.Integration of computer vision in augmented reality (AR) and virtual reality (VR) systems.Use of computer vision in medical imaging for disease diagnosis.

2020-2025 (Predicted) | Further advancements in real-time object detection and image recognition.More sophisticated use of computer vision in autonomous vehicles.Increased use of computer vision in healthcare for early disease detection and treatment.Integration of computer vision in more consumer products, like smart home devices.

Evolution of Computer Vision

Time Period

Evolution of Computer Vision

Evolution of Computer Vision

Evolution of Computer Vision

- Development of deep learning algorithms for. recognition image.

- Introduction of convolutional neural networks (CNNs) for image classification.
- Use of computer vision in autonomous vehicles for object detection and navigation.

Evolution of Computer Vision

- Advancements in real-time object detection with systems like YOLO (You Only Look Once).
- in facial recognition technology, used in various applications like unlocking smartphones and surveillance.
- Integration of computer vision in augmented reality (AR) and virtual reality (VR) systems.
- Use of computer vision in medical imaging for disease diagnosis.

Evolution of Computer Vision

- Further advancements in real-time object detection and image recognition.
- More sophisticated use of computer vision in autonomous vehicles.
- Increased use of computer vision in healthcare for early disease detection and treatment.
- Integration of computer vision in more consumer products, like smart home devices.

Applications of Computer Vision

- Healthcare: Computer vision is used in medical imaging to detect diseases and abnormalities. It helps in analyzing X-rays, MRIs, and other scans to provide accurate diagnoses.
- Automotive Industry: In self-driving cars, computer vision is used for object detection, lane keeping, and traffic sign recognition. It helps in making autonomous driving safe and efficient.
- Retail: Computer vision is used in retail for inventory management, theft prevention, and customer behaviour analysis. It can track products on shelves and monitor customer movements.
- Agriculture: In agriculture, computer vision is used for crop monitoring and disease detection. It helps in identifying unhealthy plants and areas that need more attention.
- Manufacturing: Computer vision is used in quality control in defect detect can It. manufacturing products that are hard to spot with the human eye.
- Security and Surveillance: Computer vision is used in security cameras to detect suspicious

activities, recognize faces, and track objects. It can alert security personnel when it detects a threat.

- Augmented and Virtual Reality: In AR and VR, computer vision is used to track the user's movements and interact with the virtual environment. It helps in creating a more immersive experience.

- Social Media: Computer vision is used in social media for image recognition. It can identify objects, places, and people in images and provide relevant tags.

- Drones: In drones, computer vision is used for navigation and object tracking. It helps in avoiding obstacles and tracking targets.

- Sports: In sports, computer vision is used for player tracking, game analysis, and highlight generation. It can track the movements of players and the ball to provide insightful statistics.

What is OpenCV in computer vision?

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Is cv2 and OpenCV same?

No, Actually cv2 was a old Interface of old OpenCV versions named as cv. it is the name that openCV developers choose when they created the binding generators.

Which algorithm OpenCV uses?

OpenCV uses various algorithms, including but not limited to, Haar cascades, SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), and ORB (Oriented FAST and Rotated BRIEF).

Which algorithm OpenCV uses?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Which algorithm OpenCV uses?

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund.

What more motivation do you need? Start the challenge right away!

Which algorithm OpenCV uses?

- AI-ML-DS
- Computer Vision
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++

- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter

- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design

- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases

- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is Generative AI?

-
-
-

What is Generative AI?

Generative artificial intelligence, often called generative AI or gen AI, is a type of AI that can create new content like conversations, stories, images, videos, and music. It can learn about different topics such as languages, programming, art, science, and more, and use this knowledge to solve new problems.

What is Generative AI?

For example: It can learn about popular design styles and create a unique logo for a brand or an organisation.

What is Generative AI?

Businesses can use generative AI in many ways, like building chatbots, creating media, designing products, and coming up with new ideas.

Evolution of Generative AI

Generative AI has come a long way from its early beginnings. Here's how it has evolved over time, step by step:

1.The Early Days: Rule-Based Systems

- AI systems followed strict rules written by humans to produce results. These systems could only do what they were programmed for and couldn't learn or adapt.
- For example, a program could create simple shapes but couldn't draw something creative like a landscape.

2.Introduction of Machine Learning (1990s-2000s)

- AI started using machine learning, which allowed it to learn from data instead of just following

rules. The AI was fed large datasets (e.g., pictures of animals), and it learned to identify patterns and make predictions.

- Example: AI could now recognize a dog in a picture, but it still couldn't create a picture of a dog on its own.

3.The Rise of Deep Learning (2010s)

- Deep learning improved AI significantly by using neural networks, which mimic how the human brain works. AI could now process much more complex data, like thousands of photos, and start generating new content.

- Example: AI could now create a realistic drawing of a dog by learning from millions of dog photos.

4.Generative Adversarial Networks (2014)

- GANs, introduced in 2014, use two AI systems that work together: one generates new content, and the other checks if it looks real. This made generative AI much better at creating realistic images, videos, and sounds.

- Example: GANs can create life like images of people who don't exist or filters (used in apps like FaceApp or Snapchat).

5.Large Language Models (LLMs) and Beyond (2020s)

- Models like GPT-3 and GPT-4 can understand and generate human-like text. They are trained on massive amounts of data from books, websites, and other sources. AI can now hold conversations, write essays, generate code, and much more.

- Example: ChatGPT can help you draft an email, write a poem, or even solve problems.

6.Multimodal Generative AI (Present)

- New AI models can handle multiple types of data at once—text, images, audio, and video. This allows AI to create content that combines different formats.

- Example: AI can take a written description and turn it into an animated video or a song with the help of different models integrating together.

Types of Generative AI Models

Generative AI is versatile, with different models designed for specific tasks. Here are some types:

Types of Generative AI Models

- Text-to-Text: These models generate meaningful and coherent text based on input text. They are widely used for tasks like drafting emails, summarizing lengthy documents, translating languages, or even writing creative content. Tools like ChatGPT is brilliant at understanding context and producing human-like responses.
- Text-to-Image: This involves generating realistic images from descriptive text. For Example, tools like DALL-E 2 can create a custom digital image based on prompts such as ?A peaceful beach with palm trees during a beautiful sunset,? offering endless possibilities for designers, artists, and marketers.
- Image-to-Image: These models enhance or transform images based on input image . For example, they can convert a daytime photo into a night time scene, apply artistic filters, or refine low-resolution images into high-quality visuals.
- Image-to-Text: AI tools analyze and describe the content of images in text form. This technology is especially beneficial for accessibility, helping visually impaired individuals understand visual content through detailed captions.
- Speech-to-Text: This application converts spoken words into written text. It powers virtual assistants like Siri, transcription software, and automated subtitles, making it a vital tool for communication, accessibility, and documentation.
- Text-to-Audio: Generative AI can create music, sound effects, or audio narrations from textual prompts. This empowers creators to explore new soundscapes and compose unique auditory experiences tailored to specific themes or moods.
- Text-to-Video: These models allow users to generate video content by describing their ideas in text. For example, a marketer could input a vision for a promotional video, and the AI generates visuals and animations, streamlining content creation.

- Multimodal AI: These systems integrate multiple input and output formats, like text, images, and audio, into a unified interface. For instance, an educational platform could let students ask questions via text and receive answers as interactive visuals or audio explanations, enhancing learning experiences.

Relationship Between Humans and Generative AI

In today's world, Generative AI has become a trusted best friend for humans, working alongside us to achieve incredible things. Imagine a painter creating a masterpiece, while they focus on the vision, Generative AI acts as their assistant, mixing colors, suggesting designs, or even sketching ideas. The painter remains in control, but the AI makes the process faster and more exciting.

Relationship Between Humans and Generative AI

This partnership is like having a friend who's always ready to help. A writer stuck on the opening line of a story can turn to Generative AI for suggestions that spark creativity. A business owner without design skills can rely on AI to draft a sleek website or marketing materials. Even students can use AI to better understand complex topics by generating easy-to-grasp explanations or visual aids.

Relationship Between Humans and Generative AI

Generative AI is not here to replace humans but to empower them. It takes on repetitive tasks, offers endless possibilities, and helps people achieve results they might not have imagined alone. At the same time, humans bring their intuition, creativity, and ethical judgment, ensuring the AI's contributions are meaningful and responsible.

Relationship Between Humans and Generative AI

In this era, Generative AI truly feels like a best friend—always there to support, enhance, and inspire us while letting us stay in charge. Together, humans and AI make an unbeatable team, achieving more than ever before.

Generative AI Vs AI

Criteria | Generative AI | Artificial Intelligence

Purpose | It is designed to produce new content or data | Designed for a wide range of tasks but not limited to generation

Application | Art creation, text generation, video synthesis, and so on | Data analysis, predictions, automation, robotics, etc

Learning | Uses Unsupervised learning or reinforcement learning | Can use supervised, semi-supervised, or reinforcement

Outcome | New or original output is created | Can produce an answer and make a decision, classify, data, etc.

Complexity | It requires a complex model like GANs | It has ranged from simple linear regression to complex neural networks

Data Requirement | Required a large amount of data to produce results of high-quality data | Data requirements may vary; some need little data, and some need vast amounts

Interactivity | Can be interactive, responding to user input | Might not always be interactive, depending on the application

Generative AI Vs AI

Criteria

Generative AI Vs AI

Generative AI

Generative AI Vs AI

Artificial Intelligence

Generative AI Vs AI

Purpose

Generative AI Vs AI

It is designed to produce new content or data

Generative AI Vs AI

Designed for a wide range of tasks but not limited to generation

Generative AI Vs AI

Application

Generative AI Vs AI

Art creation, text generation, video synthesis, and so on

Generative AI Vs AI

Data analysis, predictions, automation, robotics, etc

Generative AI Vs AI

Learning

Generative AI Vs AI

Uses Unsupervised learning or reinforcement learning

Generative AI Vs AI

Can use supervised, semi-supervised, or reinforcement

Generative AI Vs AI

Outcome

Generative AI Vs AI

New or original output is created

Generative AI Vs AI

Can produce an answer and make a decision, classify, data, etc.

Generative AI Vs AI

Complexity

Generative AI Vs AI

It requires a complex model like GANs

Generative AI Vs AI

It has ranged from simple linear regression to complex neural networks

Generative AI Vs AI

Data Requirement

Generative AI Vs AI

Required a large amount of data to produce results of high-quality data

Generative AI Vs AI

Data requirements may vary; some need little data, and some need vast amounts

Generative AI Vs AI

Interactivity

Generative AI Vs AI

Can be interactive, responding to user input

Generative AI Vs AI

Might not always be interactive, depending on the application

Benefits of Generative AI

Generative AI offers innovative tools that enhance creativity, efficiency, and personalization across various fields.

Benefits of Generative AI

- **Enhances Creativity:** Generative AI enables the creation of original content like images, music, and text, helping artists, designers, and writers explore fresh ideas. It bridges the gap between human creativity and machine-generated innovation, making the creative process more dynamic.
- **Accelerates Research and Development:** In fields like science and technology, Generative AI reduces the time needed for research by generating multiple outcomes and predictions, such as molecular structures in drug development. This speeds up innovation and helps solve complex problems efficiently.
- **Improves Personalization:** Generative AI creates tailored content based on user preferences. From personalized product designs to customized marketing campaigns, it enhances user engagement and satisfaction by delivering exactly what users need or want.
- **Empowers Non-Experts:** Even users without expertise can create high-quality content using Generative AI. This helps individuals learn new skills, access creative tools, and open doors to personal and professional growth.
- **Drives Economic Growth:** Generative AI introduces new roles and opportunities by fostering innovation, automating tasks, and enhancing productivity. This leads to economic expansion and the creation of jobs in emerging fields.

Limitations of Generative AI

While Generative AI offers many benefits, it also comes with certain limitations that need to be addressed

Limitations of Generative AI

- **Data Dependence:** The accuracy and quality of Generative AI outputs depend entirely on the data it is trained on. If the training data is biased, incomplete, or inaccurate, the generated content will reflect these flaws.
- **Limited Control Over Outputs:** Generative AI can produce unexpected or irrelevant results, making it challenging to control the content and ensure it aligns with specific user requirements.
- **High Computational Requirements:** Training and running Generative AI models demand significant

computing power, which can be costly and resource-intensive. This limits accessibility for smaller organizations or individuals.

- Ethical and Legal Concerns: Generative AI can be misused to create harmful content, like deepfakes or fake news, which can spread misinformation or violate privacy. These ethical and legal challenges require careful regulation and oversight to prevent abuse.

Generative AI ? FAQs

Q1. Is generative AI replacing jobs?

Generative AI ? FAQs

Generative AI isn't about replacing jobs but transforming them. It automates repetitive tasks, allowing people to focus on more creative and strategic aspects of their work. For example, content writers can use AI for inspiration or to speed up first drafts, while designers can use it to generate quick mockups.

Generative AI ? FAQs

Q2. How does Generative AI work?

Generative AI ? FAQs

Generative AI works by teaching computer programs (like GPT-3 or GANs) from lots of examples. These programs learn how things are usually done from the data they study. Then, they can use this knowledge to create new stuff when given a starting point or a request.

Generative AI ? FAQs

Q3. What are common use cases for Generative AI?

Generative AI ? FAQs

Generative AI has a wide range of applications, including content generation, language translation, chatbots, image and video creation, data augmentation, and personalized marketing. It can also be used in artistic creation, medical image generation, and more.

Generative AI ? FAQs

Q4. Is Generative AI different from other AI types?

Generative AI ? FAQs

Yes, Generative AI is different from other AI types, like classification or regression models. While those models make predictions or classify data, generative models focus on creating new, original data based on the patterns they've learned. They are versatile and used for creative tasks.

Generative AI ? FAQs

Q5. How can I get started with generative AI?

Generative AI ? FAQs

You can start by exploring tools and platforms like ChatGPT for text generation, DALL-E for image generation, or similar tools for your needs. Many platforms also provide APIs, allowing developers to integrate AI capabilities into their own applications. Learning basic prompt engineering can also help you get the most out of these tools.

Generative AI ? FAQs

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Generative AI ? FAQs

Master Machine Learning, Data Science & AI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Generative AI ? FAQs

- AI-ML-DS
- Artificial Intelligence
- Generative AI

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python

- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP

- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems

- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner

- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming

- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Getting Started with Chat GPT Tutorial

- What is ChatGPT: Everything That You Need to Know
- Getting Started With ChatGPT: A Complete Guide With Examples
- Chat GPT-4 - All You Need To Know in 2024
- GPT 4 vs GPT 3: Top Differences That You Should Know in 2025

Prompt Engineering and ChatGPT

- What is Prompt Engineering - Meaning, Working, Techniques
- 20 Best ChatGPT Prompts For Students

ChatGPT for Developers

- Roadmap of Becoming a Prompt Engineer
- Top 20 ChatGPT Prompts For Software Developers
- 15 ChatGPT Prompts For Web Developers
- 15 Must Try ChatGPT Prompts For Data Scientists
- Top 20 ChatGPT Prompts For Machine Learning
- 10 ChatGPT Prompts For UI/UX Designers
- ChatGPT Prompt to get Datasets for Machine Learning
- 10 Best Ways Developers Can Use ChatGPT-4
- How ChatGPT is Transforming the Software Development Process?

How to Use ChatGPT

- How to Use ChatGPT 4 for Free in 2025 (6 Easy Ways)
- How to Login into ChatGPT: Step-By-Step Guide
- How to Use ChatGPT API in Python?
- How To Implement ChatGPT In Django
- How to use ChatGPT to Prepare for Technical Interviews?
- How to use Chat-GPT to solve Coding Problems?
- How to Use ChatGPT to Complete Your Coding Assignments?
- How to Build a To Do App With ChatGPT?
- How to Create Your Own ChatGPT Plugin?
- How to build a chatbot using ChatGPT?
- How to Use chatgpt on Linux
- How to Use ChatGPT For Making PPT?
- How to Use ChatGPT to Write Excel Formulas
- How to Use ChatGPT to Make Your Resume?
- How To Use Chat GPT For Market Research

- How to Use the ChatGPT Chrome Extension to Write Tweets?

ChatGPT Tips and Tricks

- 10 Ways to Make Money with ChatGPT
- 10 Best Ways Youtubers Can Use ChatGPT
- 10 Best ChatGPT Plugins You Should Use
- 10 Best ChatGPT Alternatives in 2024
- 15 Best ChatGPT Chrome Extensions [2025 Updated]
- Top 7 ChatGPT R Programming Tools in 2023
- 5 Free GPT-4 Apps to Use as ChatGPT Alternatives
- ChatGPT For Business: Tips and Tricks For Success
- Creating ChatGPT Clone in Python
- Generate Images With OpenAI in Python

ChatGPT Competitors

- ChatGPT vs Google BARD - Top Differences That You Should Know
- AI Chatbots: ChatGPT vs. Bing vs. Bard

How to Earn with Chatgpt

- 10 Ways to Make Money with ChatGPT
- How to Earn Money with ChatGPT (Student Edition)
- How to Make Money with ChatGPT (WFH Edition)
- How to Earn \$10000 Per Month Through Side Gigs : Answered by ChatGPT
- How to use ChatGPT for passive income?
- How to Make Money With AI in 2024 - Secrets Revealed

ChatGPT blogs

- ChatGPT: 7 IT Jobs That AI Can't Replace
- Jobs That ChatGPT Can Replace in Near Future

- ChatGPT Impact on the Global Job Market
- Is ChatGPT and Layoffs are Related?
- How ChatGPT is Transforming the Software Development Process?
- Level up your ChatGPT Game with OpenAI's Free Course on Prompt Engineering for Developers

What is ChatGPT: Everything That You Need to Know

-
-
-

What is ChatGPT: Everything That You Need to Know

In the era of Artificial intelligence, ChatGPT has emerged as a prominent player, revolutionizing the way we interact with machines. With its advanced language capabilities and natural conversation flow, ChatGPT has captured the imagination of users worldwide.

What is ChatGPT: Everything That You Need to Know

ChatGPT

What is ChatGPT: Everything That You Need to Know

In this comprehensive blog post, we'll dive into everything you need to know about ChatGPT ? its definition, capabilities, applications, limitations, and impact on various industries.

What is ChatGPT: Everything That You Need to Know

Table of Content

What is ChatGPT: Everything That You Need to Know

- What is ChatGPT?
- How Does ChatGPT Work?
- Who made ChatGPT?
- How Much Does ChatGPT Cost?

- Is ChatGPT a Reliable Technology?
- How to Use ChatGPT?
- What is CHATGPT Prompt Engineering?
- Step-By-Step Process of Using ChatGPT
- Limitations of ChatGPT:
- How People are using ChatGPT?
- Can ChatGPT Generate a Programming Language Code/Program?
- Can ChatGPT Generate Algorithms?
- Useful Resources

What is ChatGPT?

ChatGPT is a product of OpenAI, a leading research organization focused on artificial intelligence. It is built upon the GPT (Generative Pre-trained Transformer) architecture, a state-of-the-art model for natural language processing (NLP). Developed by a team of researchers and engineers, ChatGPT leverages large-scale training data to generate human-like responses to user input.

What is ChatGPT?

It has been aimed to provide answers based on feelings, auto-detection of words, and provide outputs based on the inputs given by the users.

What is ChatGPT?

Sooner, this system will go out on all major platforms and so the accuracy will be tested on all different forms. The company also aims to target all mainstream to replace them with traditional chatbot methods.

How Does ChatGPT Work?

ChatGPT employs deep learning techniques to understand and generate text-based responses in natural language. It utilizes a Transformer architecture, which allows it to analyze and process large amounts of text data, learn patterns and correlations, and generate contextually relevant

responses. Through continuous training and refinement, ChatGPT improves its language understanding and response generation capabilities over time

How Does ChatGPT Work?

In other words, it is nothing more than a model that is being trained by humans and powered by AI, and based on the inputs and feedback, it shifts its pattern and responds accordingly. OpenAI stated in a statement that this model works on Reinforcement Learning from Human Feedback or RLHF that can reject inappropriate inputs, train themselves to provide the best solution, and answer all the users' queries.

Who made ChatGPT?

OpenAI, an AI and research company created ChatGPT. The company launched ChatGPT on November 30, 2022. The ChatGPT app utilizes advanced AI algorithms to engage users in conversations, offering solutions and assistance familiar to human interaction.

How Much Does ChatGPT Cost?

Till now ChatGPT is free to use for everyone, regardless of what you use it for, including writing, coding, and much more, there are no limitations but sometimes it may show some heavy traffic due to which it might not function properly.

How Much Does ChatGPT Cost?

There is a subscription option that users can take advantage of that costs \$20/month. The paid subscription model guarantees users extra perks, such as general access even at capacity, faster response times, and access to the internet through plugins.

Is ChatGPT a Reliable Technology?

The initial version had some drawbacks which have been identified and rectified (based on human feedback), and version 2 came into play but is still considered less accurate. Not only this, in fact when a user will start this application, a disclaimer warning will pop up on their screen which will

warn the users that ChatGPT is not 100% accurate and can also provide answers that might discomfort the users.

Is ChatGPT a Reliable Technology?

Being its unbiased nature, it fetches the queries and passes on the messages to the end users that might be misleading to their readers. But the brighter side is that it works as a front-responder of any business/brand and can provide outputs that can be a much helpful solution to their desired customers.

What is CHATGPT Prompt Engineering?

ChatGPT prompt engineering means crafting questions or messages in a clever way to get the best answers from the AI. It's like asking the right questions to a friend to get the most helpful response. This technique helps make ChatGPT conversations more useful and accurate.

Step-By-Step Process of Using ChatGPT

You can log in to ChatGPT within a few clicks, refer to the below step guidance for best reference:

STEP 1: Open the official website then click on TRY CHATGPT & your chatgpt sign up

ChatGPT Landing Page

STEP 1: Open the official website then click on TRY CHATGPT & your chatgpt sign up

STEP 2: A new window will then appear on your screen and you'll be asked to LOG IN

ChatGPT Log In And Sign Up Buttons

STEP 2: A new window will then appear on your screen and you'll be asked to LOG IN

You will be able to see a screen with multiple options that you can do with ChatGPT,

ChatGPT interface

STEP 4: We have shared the screenshot for reference that how you can interact using

STEP 4: We have shared the screenshot for reference that how you can interact using

This clearly indicates how interactive ChatGPT could be and can help you out with all the possible suggestions. For instance, you can also ask to translate any language, write lyrics, and much more.

STEP 4: We have shared the screenshot for reference that how you can interact using

Note: Since, it's still under the development phase so if in case you find any issue with the provided output, you can submit your feedback and you can also hit the like and dislike button (available on the right side of every ChatGPT response.)

Limitations of ChatGPT:

The list of limitations is long when using ChatGPT, however, we have focused on providing concrete differences that can be fixed in the upcoming future. Have a look at some of the limitations below:

Limitations of ChatGPT:

- No real-time output
- External support from any URL (for reference) is not allowed
- The location-based facility is still missing
- There are cases where ChatGPT was unable to answer basicNo real-time output
- External support from any URL (for reference) is not allowed riddles, have a look at the screenshot for the best reference
- Lack of understanding Context
- Vulnerability to misinformation

Limitations of ChatGPT:

Human Alike Response from ChatGPT

Limitations of ChatGPT:

In most cases, it has been observed that ChatGPT is lagging behind and needs improvements. Till,

that time, the company is targeting to work towards making their AI-based system more convenient.

How People are using ChatGPT?

It took 4 days for ChatGPT to hit a million user milestone which is comparatively high for many popular applications that we use today. For example, Instagram took near about 90 days to reach that milestone. The charm of this technology has forced people to try out things in different ways and the Internet is flooded with the things that people have been doing with this latest sensation. Let's look at some of them and analyze how to use a chatbot:

How People are using ChatGPT?

- Coding: The days are not far away when technologies like this would change the meaning of programming and people have been enjoying working on different languages using this tool.
- Poetry: Many users have posted updates where they have shared their experiences of writing poems using ChatGPT
- Song Writing: Since this tool works with AI-model and is capable of generating texts, it can easily write a song in a matter of time.
- Game/Movie Suggestion: It is also being considered as a replacement for the Google search engine which is actually a big statement but the wonders it is doing are still unimaginable and based on users' inputs, it can provide suggestions for food, movies, games, etc.

Can ChatGPT Generate a Programming Language Code/Program?

Yes, ChatGPT can generate code in various programming languages such as Python, Java, C++, and more. However, it's important to note that the quality of the code generated may vary and it may not always be free of errors or optimal for the intended task.

Can ChatGPT Generate a Programming Language Code/Program?

ChatGPT Generated Program for prime numbers

Can ChatGPT Generate Algorithms?

Yes, ChatGPT can generate algorithms for various tasks such as sorting, searching, and mathematical operations. However, the quality and efficiency of the algorithms generated may vary, so it's important to validate and test them before use.

Can ChatGPT Generate Algorithms?

ChatGPT generated Algorithm

Useful Resources

- Chat GPT-4 ? All You Need To Know
- How to Use ChatGPT 4 For Free?
- How to Use ChatGPT API in Python?
- 10 Best ChatGPT Alternatives in 2023 (Free and Paid)

Conclusion

In conclusion, ChatGPT represents a major milestone in the field of artificial intelligence, showcasing the potential of advanced NLP models to understand and generate human-like text. With its impressive capabilities and broad range of applications, ChatGPT is poised to continue shaping the future of human-machine interaction and driving innovation across industries. As ChatGPT evolves and matures, it will undoubtedly play an increasingly integral role in our daily lives, revolutionizing how we communicate, learn, and interact with technology.

What is the full form of ChatGPT?

The full form of ChatGPT is ?Chat Generative Pre-training Transformer.?

Is ChatGPT free to use?

Since it is still in the development phase and their team is continuously working to make ChatGPT more smooth and interactive so currently it is free to use and open to all. However, the company has not disclosed any future plans regarding the usage.

Will ChatGPT replace Google?

The answer to this is ?NO?, ChatGPT is an AI-based trained model that works on generating output based on human interaction (provided inputs) and there are times when ChatGPT is not even answering the relevant answers. On the other hand, Google is an internationally established brand serving as a home brand for search engines for every third person in the world. Google offers many other services that are more precise, vast, and accurate. So, ChatGPT is likely not going to replace Google completely.

Who owns Open AI?

Open AI is an AI-based research lab that has two different verticals, i.e.

Who owns Open AI?

- OpenAI LP
- OpenAI INC

Who owns Open AI?

The organization was founded by Elon Musk, Ilya Sutskever, Wojciech Zaremba Greg Brockman, and Sam Altman.

What is ChatGPT-4?

ChatGPT-4 is the latest version of the ChatGPT series, developed by OpenAI. It's an advanced AI-powered chatbot that excels in conversational interactions. With improved capabilities and enhanced features, ChatGPT-4 offers more accurate and natural responses, making it a cutting-edge tool for various applications in communication, content generation, and problem-solving.

Will ChatGPT take jobs?

The concept of ChatGPT is so unique that it gained popularity in a matter of no time. But, since it's in a new phase and there are tons of work required to generate quality output. So, the answer is ?NO? it's not going to take jobs in the upcoming years.

What are ChatGPT Alternatives?

Since its launch, many companies have come out offering different tools for AI-based content, among which are listed below:

What are ChatGPT Alternatives?

- Bloom
- Replika
- Jasper
- FaceApp, etc.

What are ChatGPT Alternatives?

What are ChatGPT Alternatives?

Get 90% Course fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today. The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

What are ChatGPT Alternatives?

- GBlog
- AI Chatbot
- ChatGPT

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy

- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Compute Services

- What is Elastic Compute Cloud (EC2)?
- AWS Lambda
- Amazon Elastic Container Service (ECS)

Storage Services

- Introduction to AWS Simple Storage Service (AWS S3)
- Introduction to AWS Elastic Block Store(EBS)
- Amazon RDS - Introduction to Amazon Relational Database System

AWS Networking Services

- Amazon VPC - Introduction to Amazon Virtual Private Cloud
- Introduction to Amazon Route53
- Amazon VPC - Working with Direct Connect Service

AWS Database Services

- AWS DynamoDB - Creating a Table
- What is Amazon Aurora? In Detail Explanation
- Difference between Amazon Redshift and PostgreSQL

AWS Machine Learning Services

- What is SageMaker in AWS?
- What is AWS Deeplens?

AWS Developer Tools

- Working with AWS CodeCommit
- How to Build a CI/CD Pipeline with AWS?
- How to Build a AWS Data Pipeline?

AWS Management and Monitoring

- Introduction to Amazon CloudWatch
- AWS CloudTrail

AWS Getting Hands-On

- Launch the EC2 Instance in AWS with New Instance Wizard
- Amazon S3 - Creating a S3 Bucket
- Build Serverless Web App with AWS Lambda
- How to Install AWS CLI on Ubuntu?

Security and Identity

- Identity and Access Management (IAM) in Amazon Web Services (AWS)
- Amazon Web Services - Managing Invalid Keys in Key Management System
- Understanding the AWS Cloud Security Model

Cost Management

- AWS Cost Explorer
- Amazon Web Services Budget Setup

Deployment and Automation

- What is AWS Cloudformation?
- Introduction to AWS Elastic Beanstalk

Advanced Topics

- Serverless Computing
- How Amazon Uses Machine Learning?

Certification and Training

- AWS Certification Path - Levels, Exam, Cost
- How to Prepare for AWS Certification Exam?
- Explore AWS Cloud Practitioner Certification Course!

Amazon Web Services (AWS) Tutorial

-
-
-

Amazon Web Services (AWS) Tutorial

This AWS tutorial, or Amazon Web Service tutorial, is designed for beginners and professionals to learn AWS's basic and advanced concepts. Learn about the various topics of AWS such as introduction, history of AWS, global infrastructure, features of AWS, IAM, storage services, database services, application Services, etc., and other AWS products such as S3, EC2, Lambda, and more. By the end of this tutorial, readers will have a basic understanding of what AWS is and how it can be used to support their computing needs.

Amazon Web Services (AWS) Tutorial

Amazon Web Services (AWS) Tutorial

AWS or Amazon Web Services, is a cloud computing platform that offers on-demand computing services such as virtual servers and storage that can be used to build and run applications and websites. AWS is known for its security, reliability, and flexibility, which makes it a popular choice for organizations that need to store and process sensitive data.

What is Amazon Web Service or AWS

Amazon Web Services (AWS) is a cloud computing platform offered by Amazon. It provides a wide range of on-demand services like computing power, storage, and databases, allowing businesses to scale and manage their IT resources efficiently. AWS offers services such as EC2 for virtual servers, S3 for scalable storage, RDS for managed databases, and Lambda for serverless computing. By using AWS, companies can reduce infrastructure costs, improve flexibility, and deploy applications globally with ease.

Prerequisites to Learn AWS

Before jumping to the AWS Tutorial, it's recommended to have a basic foundational understanding of operating systems, computer networking, basic coding commands in Linux terminals, and some prior knowledge of cloud computing.

AWS Tutorial ? Table of Content

- Introduction to AWS
- AWS IAM ? Identity and Access Management
- Computing in AWS
- Section 1 ? AWS EC2 ? Backbone of AWS
- Section 2 ? AWS EBS ? Elastic Bean Stalk
- Section 3 ? AWS EBS ? Elastic Block Store
- Section 4- AWS AMI
- Section 5 ? AWS Load Balancer
- Section 6 ? AWS Lambda
- Section 7 ? AWS CloudWatch
- Section 8 ? AWS AutoScaling
- Section 9 ? Other Elastic Computes (ECS, EKS, ECR)
- AWS Storage Services
- AWS Application Services
- AWS Database Services
- AWS VPC ? Virtual Private Cloud
- AWS Billing and Management
- Machine Learning and IoT
- Advantages of AWS
- Application of AWS

AWS Tutorial ? Table of Content

- Section 1 ? AWS EC2 ? Backbone of AWS
- Section 2 ? AWS EBS ? Elastic Bean Stalk
- Section 3 ? AWS EBS ? Elastic Block Store
- Section 4- AWS AMI
- Section 5 ? AWS Load Balancer
- Section 6 ? AWS Lambda
- Section 7 ? AWS CloudWatch
- Section 8 ? AWS AutoScaling
- Section 9 ? Other Elastic Computes (ECS, EKS, ECR)

Introduction to AWS

In this section, we are going to describe all about the AWS, like its definition, history, setup and more. So, explore this section to get an introduction of Amazon Web Service.

Introduction to AWS

- Definition of AWS
- History of AWS
- Setup AWS Business Account
- Setup Free Tier Account

AWS IAM ? Identity and Access Management

AWS Identity and Access Management (IAM) is a crucial component of the Amazon Web Services (AWS) ecosystem, providing centralized control over user access to AWS resources. With IAM, administrators can securely manage user identities, assign permissions, and control privileges across the AWS environment.

AWS IAM ? Identity and Access Management

- AWS IAM
- AWS SAML

- IAM Identities
- IAM Roles
- Creating IAM roles for EC2
- Denying Access using IAM policy for EC2 and EBS Instance

Computing in AWS

Computing in AWS? refers to the utilization of Amazon Web Services (AWS) for various computational tasks and workloads. Here in this section we have listed all the topics AWS.

Section 1 ? AWS EC2 ? Backbone of AWS

- What is EC2
- How to create an Instance in EC2
- Types of EC2
- Price Model for EC2
- Introduction to EC2 Spot Instances
- Create a Windows EC2 Instance and Connect Using RDP
- Create an EC2 Instance User Data Script
- Flexibility in EC2 Spot Instances
- Rules for Spot Instances
- Installation of Python3 on AWS EC2?
- Installation of GO on EC2?
- Connecting EC2 using Bastion Host

Section 2? AWS EBS ? Elastic Bean Stalk

- What is EBS (elastic Bean Stalk)
- How to Attach EBS Volume in EC2
- Replacing Unhealthy EC2 Instances in EBS
- Launching an Application on AWS Beanstalk

- Add Security Group in EBS

Section 3 ? AWS EBS ? Elastic Block Store

- Introduction to EBS
- EBS Snapshot
- Replace EC2 in EBS Environment
- Change EBS Encryption Key

Section 4- AWS AMI

- Definition of AMI
- Creation of AMI

Section 5 ? AWS Load Balancer

- Definition of AWS Load Balancer
- Creation of Load Balancer

Section 6 ? AWS Lambda

- AWS Lambda
- Create an AWS Lambda Function
- Insert data in DynamoDB using Lambda

Section 7 ? AWS CloudWatch

- AWS CloudWatch
- AWS Cloudwatch synthetics
- Difference between CloudWatch and CloudTrail
- Bash Scripting

Section 8 ? AWS AutoScaling

- Scaling in EC2
- Create AutoScaling Group in EC2

Section 9 ? Other Elastic Computes (ECS, EKS, ECR)

- What is ECS
- What is EKS
- Generating Log Bundle for EKS Instance
- What is ECR
- Difference between EBS and EFS

AWS Storage Services

- AWS S3
- AWS S3 Storage types/classes
- AWS S3 versioning
- Creation of AWS S3 bucket
- Difference between Amazon S3 and Box
- Difference between Amazon S3 and TitanFile
- Difference between Amazon S3 and SecureSafe
- AWS S3 Lifecycle Management
- Cross Region Replication
- Creation of Cross Region Replication
- AWS Glacier
- How to setup AWS Backup
- Disaster Recovery Strategies
- AWS EBS (elastic block Store)
- AWS EFS
- Difference between EBS and EFS
- Difference between S3 and EBS
- Cross Region Replication
- Cloud Front CDN

- Creating Cloud Front CDN
- Storage Gateway
- AWS Snowball
- S3 Transfer Acceleration

AWS Application Services

- AWS SQS
- AWS SWF
- AWS SNS
- Installation of AWS SNS
- Elastic Transcoder
- API Gateway
- AWS Kinesis
- AWS Fargate

AWS Database Services

- Difference between Relational and Non-Relational Database
- AWS DynamoDB
- Working with DynamoDB Tables
- Introduction to NoSQL Workbench
- Creating a table in AWS DynamoDB
- Querying in a table in AWS DynamoDB
- Working with Queries
- Working with backups
- Web-based Setup
- AWS Aurora
- Difference between Aurora and Redshift
- Working with Third-Party Data in Redshift

- Copy an Amazon Redshift Cluster to a Different AWS Account
- Introduction to AWS RDS
- Creating RDS Instance
- What is ElasticCache

AWS VPC ? Virtual Private Cloud

- AWS VPC
- Security in VPC
- VPC Peering
- Creating your own custom VPC
- VPC Networking Components
- How to check metadata about VPC
- Working with VPC and subnets
- Direct Connect
- NAT Gateways
- How VPC interacts with Direct Connect Services
- AWS VPC Endpoints
- AWS VPC FlowLogs
- AWS Bastion Host
- Connect an EC2 of Private Subnet using Bastion Host
- NACL vs Security Group
- AWS Lightsail
- Launch a WordPress using lightsail
- Resolve Domain verification using SES
- AWS CloudFormation
- Unregistered Function Error in Athena
- What is AWS Amplify

- Creating a User Pool in AWS Cognito
- What is DNS
- What is Route53
- Removing Restrictions from Port 25 in EC2
- AWS Data Pipeline

AWS Billing and Management

- Introduction to Cost Explorer and Cost Management
- AWS Budget Setup
- Cost Usage and Report

Machine Learning and IoT

- AWS Sagemaker
- Using Custom UI in Sagemaker
- IoT Products

AWS Interview Questions

AWS Interview Question latest ? 2023

Advantages of AWS

Here are Some Advantages of Amazon Web Services are ?

Advantages of AWS

- Cost Efficient ?AWS does not necessitate any upfront investment, long-term commitment, or minimum expense for the setup of Cloud Infrastructure.
- Flexibility ?AWS offers Effortless hosting of Applications and it provides instant availability of new features and technology.
- Scalability ?AWS offer Features like AutoScaling and Elastic Load Balancing and their scale Increase and Decrease according to demand or traffic on the application.

- Security ?AWS provides end-to-end encryption technology and privacy to its customers.

Application of AWS

Amazon Web Services (AWS) is being increasingly adopted by many large enterprises such as Netflix, McDonald's, Airbnb, NASA, and Samsung to expand their businesses. AWS offers a variety of applications, some of which include:

Application of AWS

- Storage and Backup
- Social Networking
- Mobile Apps
- Websites
- Gaming

Conclusion

AWS or Amazon Web Services is a powerful cloud computing platform that offers a wide range of on-demand computing services such as virtual servers and storage. This tutorial has provided a brief introduction to cloud computing and given an overview of the various AWS products and services that are available. By the end of this tutorial, readers will have a solid understanding of AWS.

How do I start learning AWS?

To start learning AWS, you can explore various online tutorials (GeeksforGeeks AWS Tutorial), official AWS documentation, and training courses offered by AWS. Additionally, you can practice by creating an AWS Free Tier account and experimenting with different services.

Which AWS services should I focus on as a beginner?

As a beginner, it's recommended to focus on core services such as Amazon EC2 (Elastic Compute Cloud), Amazon S3 (Simple Storage Service), and Amazon RDS (Relational Database Service).

These services form the backbone of many AWS solutions and are commonly used.

Are there any free resources for learning AWS?

Yes, AWS provides a Free Tier that offers limited access to many AWS products. This allows new users to explore and try out the services for free. Additionally, there are various free online tutorials, videos, and documentation available on the AWS website and other educational platforms.

How can I prepare for AWS certification exams?

To prepare for AWS certification exams, you can enroll in official AWS training courses, study the exam guides and whitepapers provided by AWS, and use practice tests to assess your knowledge. Hands-on experience with AWS services is also crucial for exam preparation.

What are some common challenges when working with AWS?

Common challenges when working with AWS include managing costs effectively, understanding and optimizing the complex pricing structure, ensuring security and compliance, and selecting the most suitable AWS services for specific use cases.

What are some common challenges when working with AWS?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

What are some common challenges when working with AWS?

A

What are some common challenges when working with AWS?

- Amazon Web Services

- DevOps
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas

- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science

- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

CSS Text Formatting

- CSS Text Formatting
- CSS text-align-last Property
- CSS text-decoration Property
- CSS text-decoration-color Property
- CSS text-decoration-line Property
- CSS text-decoration-style Property
- CSS text-indent Property
- CSS text-justify Property
- CSS text-overflow Property
- CSS text-transform Property
- CSS text-shadow Property
- CSS letter-spacing Property
- CSS line-height Property
- CSS direction Property
- CSS word-spacing Property

CSS Backgrounds

- CSS Background
- CSS background-color Property
- CSS background-image Property
- CSS background-attachment Property
- CSS background-position Property
- CSS background-origin property
- CSS background-clip Property

CSS Lists

- CSS Lists

- CSS list-style-type Property
- CSS list-style-image Property
- CSS list-style-position Property
- CSS list-style Property
- CSS list-style-type Property

CSS Box model

- CSS Box Model
- CSS Height and Width
- CSS Margins
- CSS Borders
- CSS border Property

CSS Positioning Elements

- CSS Positioning Elements
- Relative vs Absolute vs Fixed Position in CSS
- CSS bottom Property

CSS flexbox

- Introduction to CSS Flexbox
- CSS Flexbox and Its Properties
- CSS flex Property
- CSS flex-grow Property
- CSS flex-shrink Property
- CSS flex-basis Property
- CSS flex-wrap property

CSS Grid Layout

- CSS Grid Layout Module

- CSS column-gap Property
- CSS gap Property
- CSS grid Property
- CSS grid-area Property
- CSS grid-auto-columns Property
- CSS grid-auto-flow Property
- CSS grid-auto-rows Property
- CSS grid-column Property
- CSS grid-column-end Property
- CSS grid-column-gap Property
- CSS grid-column-start Property
- CSS grid-gap Property
- CSS grid-row Property
- CSS grid-row-end Property
- CSS grid-row-gap Property
- CSS grid-row-start Property
- CSS grid-template Property
- CSS grid-template-areas Property
- CSS grid-template-columns Property
- CSS grid-template-rows Property

Responsive Web Design

- Basics of Responsive Web Design - Media Queries
- Short note on Responsive Web Design
- Features of Responsive Web Design
- CSS Media Queries

CSS Transitions

- CSS Transitions
- Transition shorthand with multiple properties in CSS?
- CSS transition Property
- CSS transition-duration Property
- CSS transition-timing-function Property
- CSS transition-delay Property

CSS Animations

- CSS Animations
- CSS animation-name Property
- CSS animation-duration Property
- CSS animation-timing-function Property
- CSS animation-delay Property
- CSS animation-iteration-count Property
- CSS animation-direction Property
- CSS animation-fill-mode Property
- CSS animation-play-state Property

CSS Tutorial

-
-
-

CSS Tutorial

CSS stands for Cascading Style Sheets. It is a stylesheet language used to style and enhance website presentation.

CSS Tutorial

- CSS is one of the main three components of a webpage along with HTML and JavaScript.

- HTML adds Structure to a Webpage, JavaScript adds logic to it and CSS makes it visually appealing or stylish. It controls the layout of a web page i.e. how HTML elements will be displayed on a webpage.
- CSS was released (in 1996), 3 years after HTML (in 1993). The main idea behind its use is, it allows the separation of content (HTML) from presentation (CSS). This makes websites easier to maintain and more flexible.

How to Add CSS to HTML?

There are three different ways to add CSS styles to an HTML document, these are -

1. Inline CSS

Use the `style` attribute on the HTML element to add style to the web page. It is used for small projects.

2. Internal CSS

Place the CSS styles within a `<style>` tag inside the HTML file, usually inside the `<head>` section.

3. External CSS

Create a separate CSS file with a `.css` extension and link this file to your HTML file using the `<link>` tag.

3. External CSS

Introduction to Modern CSS

This section covers the basic understanding of CSS, its advantages and disadvantages.

Introduction to Modern CSS

- Introduction to CSS
- What is Cascading in CSS?
- Advantages and Disadvantages of CSS

CSS Fundamentals

This section covers the fundamental topics of CSS

CSS Fundamentals

- CSS Syntax
- CSS Ruleset
- CSS Selectors
- CSS Colors and Units
- Text Styling
- Box Model
- Positioning
- Layout Techniques
- CSS Margins, Padding, and Borders
- Responsive Design
- CSS Display Property
- CSS Inheritance
- CSS Specificity
- Backgrounds and Gradients
- Text Styling
- Positioning Methods
- Float and Clear
- Z-Index
- CSS Variables
- Opacity and Visibility
- Box Shadows and Text Shadows
- Transform Property

CSS Selectors

This Section contains all the information about all the simple selectors in CSS

CSS Selectors

- CSS Attribute Selector
- CSS Id Selector
- Class Selector
- Universal Selector
- Pseudo-Class Selector
- Pseudo-Element Selector

CSS Combinators

This Section contains all the information about combinatory selectors in CSS

CSS Combinators

- General Sibling selector (~)
- Adjacent Sibling selector (+)
- Child selector (>)
- Descendant selector (space)

CSS Layout Management

This Section contains all the information about various layouts in CSS

CSS Layout Management

- CSS Box model
- CSS Layout
- Styling Text

Styling Techniques in CSS

This Section contains all the information about various styling techniques in CSS

Styling Techniques in CSS

- CSS Fonts
- CSS Colors

- CSS Backgrounds
- CSS Borders
- CSS Grid
- CSS Flexbox
- CSS Images
- CSS Lists

Logic Implementations in CSS

This Section Covers all the mathematical logic that can be applied in CSS.

Logic Implementations in CSS

- CSS Counters
- CSS Columns
- CSS Conditional Rules
- CSS Logical Properties
- CSS Math functions

CSS Responsive Design and Media Queries

This Section contains all the designing techniques in CSS used for various use cases

CSS Responsive Design and Media Queries

- CSS Media queries
- CSS Nesting style rules
- CSS Positioning

CSS Preprocessors

This Section contains information about the preprocessors used in CSS.

CSS Preprocessors

- CSS Preprocessor SASS

- CSS Preprocessor LESS
- CSS Preprocessors LESS vs SASS

Advanced CSS Topics

This Section contains various information about advanced topics in CSS

Advanced CSS Topics

- CSS Transforms
- CSS Transitions
- CSS Animations
- CSS Variables
- CSS Properties
- CSS Function
- CSS :is selector
- CSS :where selector
- CSS shadow DOM

CSS Online Quizzes

To achieve a solid understanding of CSS, it's essential to engage with CSS quizzes and MCQs. These CSS quizzes can enhance your ability to solve similar questions and improve your problem-solving skills.

CSS Online Quizzes

Here are some quiz articles related to CSS 3:

CSS Online Quizzes

- Basic CSS Quiz
- Intermediate CSS Quiz
- Advanced CSS Quiz

CSS Practical projects for beginners

The theoretical knowledge is not enough to understand CSS. So, it is required to work on some real life projects to learn CSS easily. Working on such HTML & CSS projects will test your CSS knowledge and you will get some hands-on experience.

CSS Practical projects for beginners

Below are some HTML and CSS projects for better understanding.

CSS Practical projects for beginners

- Design Geeks for Geeks logo
- Meet the Team Page Design
- Tribute Page Design
- Design a web page
- Contact Us Page Design
- Create Browsers Window
- Design Email Newsletter

CSS Interview Preparation Questions

- Don't miss our CSS Interview Questions and Answers before going for your interview.
- CSS Cheat-Sheet for Beginners (2024) -A Basic Guide to CSS

CSS Frameworks

CSS frameworks are a collection of pre-written CSS files (and sometimes JavaScript components) that offer reusable code for common tasks such as buttons, grids, forms, and navigation menus.

CSS Frameworks

These CSS frameworks provide a set of standardized, reusable components and a predefined structure, allowing developers to create responsive and aesthetically pleasing websites with reduced effort.

CSS Frameworks

Other Resources

- CSS Complete Guide - A to Z CSS Concepts
- Learn CSS: Free CSS Course

CSS Versions

- CSS1: The foundation, released in 1996, introduced basic styling capabilities for fonts, colors, and margins.
- CSS2: Expanded in 1998, adding positioning elements, pseudo-classes, and improved layout options.
- CSS 2.1: Further refinements in 2004, including improvements to inheritance and box model properties.
- CSS3: Introduced from 2001 onwards, CSS3 isn't a single version but a collection of modules adding features like animations, media queries, and web fonts. It's constantly evolving.

Why learn CSS?

1. Enhance Visual Appeal: CSS allows you to style your web pages, making them visually appealing and engaging. Here's why it matters:

Why learn CSS?

- User Experience (UX): Well-designed websites attract and retain users. CSS enables you to create beautiful layouts, choose fonts, and apply colors that resonate with your audience.

Why learn CSS?

2. Responsive Design: In today's mobile-first world, responsive design is crucial. CSS empowers you to:

Why learn CSS?

- Media Queries: Adapt your layout based on screen size (desktop, tablet, mobile).

- Flexbox and Grid: Create flexible, adaptive designs that look great on any device.

Why learn CSS?

3. SEO Benefits: CSS indirectly impacts your site's SEO. Here's how:

Why learn CSS?

- Page Load Speed: Well-organized CSS files load faster, improving user experience. Google considers page speed as a ranking factor.
- Structured Content: Properly styled HTML (thanks to CSS) enhances readability for search engines and users.
- Mobile Friendliness: Responsive CSS ensures your site performs well on mobile devices, positively affecting rankings.

Why learn CSS?

4. Efficient Maintenance: CSS promotes clean code and separation of concerns:

Why learn CSS?

- Modularity: Separate CSS files allow easy updates without affecting other parts of your site.
- Consistency: Apply styles consistently across your site using classes and IDs.

Why learn CSS?

5. Career Opportunities: Learning CSS opens doors to various roles:

Why learn CSS?

- Front-End Developer: Mastering CSS is essential for front-end development.
- Web Designer: CSS skills are fundamental for creating stunning web layouts.
- Full-Stack Developer: Understanding CSS complements back-end skills.

What is CSS?

CSS (Cascading Style Sheets) is a stylesheet language used to control the presentation and layout

of HTML elements on a webpage. It defines styles such as colors, fonts, spacing, and positioning, enhancing the visual appearance of web content.

What is the Full Form of CSS?

CSS stands for Cascading Style Sheets. It is a style sheet language used to describe the presentation and formatting of a document written in HTML or XML.

What are the types of CSS?

There are three types of CSS: Inline CSS, which is applied directly within an HTML element; Internal CSS, written inside the `<style>` tag in the HTML document's head; and External CSS, linked through a separate .css file using the `<link>` tag.

How do I link an external CSS file to an HTML document?

Use the `<link>` element within the `<head>` section of your HTML document. Example: `<link rel="stylesheet" type="text/css" href="styles.css">`

What is the purpose of the 'box model' in CSS?

The box model is a fundamental concept in CSS that describes the layout of elements. It consists of content, padding, border, and margin, which collectively determine the size and spacing of an element.

How can I center an element horizontally and vertically in CSS?

To center horizontally, use `margin: auto;` on the element. For vertical centering, consider using Flexbox (`display: flex; align-items: center; justify-content: center;`) or Grid (`display: grid; place-items: center;`).

What is the difference between 'margin' and 'padding' in CSS?

Margin is the space outside an element, creating space between the element and its surrounding elements. Padding is the space inside an element, creating space between the element's content and its border.

How do media queries contribute to responsive design in CSS?

Media queries allow developers to apply styles based on characteristics such as screen width, height, or device orientation. They are crucial for creating responsive designs that adapt to different devices and screen sizes.

How do media queries contribute to responsive design in CSS?

Dive deep into the world of JavaScript with our JavaScript Full Course Online, offering everything from the basics to advanced techniques. Learn how to build interactive websites, optimize your code, and earn a certification to showcase your skills. Take the Three 90 Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Push your limits and stay motivated as you work towards mastering JavaScript. Join now and get started on your JavaScript learning journey!

How do media queries contribute to responsive design in CSS?

- CSS
- Web Technologies
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution

- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures

- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs

- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker

- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy

- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced

- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course

- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE

- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Data Structures Tutorial

Algorithm Tutorial

Competitive Programming

DSA Tutorial - Learn Data Structures and Algorithms

-
-
-

DSA Tutorial - Learn Data Structures and Algorithms

DSA (DataStructuresandAlgorithms) is the study of organizing data efficiently using data structures like arrays, stacks, and trees, paired with step-by-step procedures (or algorithms) to solve problems effectively. Data structures manage how data is stored and accessed, while algorithms focus on processing this data.

Why to Learn DSA?

- Learning DSA boosts your problem-solving abilities and make you a stronger programmer.
- DSA is foundation for almost every software like GPS, Search Engines, AI ChatBots, Gaming Apps, Databases, Web Applications, etc
- Top Companies like Google, Microsoft, Amazon, Apple, Meta and many other heavily focus on DSA in interviews.

How to learn DSA?

- Learn at-least one programming language (C++, Java, Python or JavaScript) and build your basic logic.
- Learn about Time and Space complexities
- Learn Data Structures and Algorithms
- Practice DSA

- Solve problems daily using GfG POTD, weekly using GfG Weekly Contest and monthly using GfG Job-A-Thon.

How to learn DSA?

Hoping you have learned a programming language of your choice, here comes the next stage of the roadmap - Learn about Time and Space Complexities.

1. Logic Building

Once you have learned basics of a programming language, it is recommended that you learn basic logic building

1. Logic Building

- Logic Building Problems
- Practice Problems on Mathematical Algorithms

2. Learn about Complexities

To analyze algorithms, we mainly measure order of growth of time or space taken in terms of input size. We do this in the worst case scenario in most of the cases. Please refer the below links for a clear understanding of these concepts.

2. Learn about Complexities

- Complexity Analysis Guide
- Quiz on Complexity Analysis of Algorithms

3. Array

Array is a linear data structure where elements are allocated contiguous memory, allowing for constant-time access.

3. Array

- Array Data Structure Guide

- Practice Problems on Arrays
- Top 50 Array Coding Problems for Interviews

4. Searching Algorithms

Searching algorithms are used to locate specific data within a large set of data. It helps find a target value within the data. There are various types of searching algorithms, each with its own approach and efficiency.

4. Searching Algorithms

- Guide on Searching Algorithms
- Practice Problems on Searching
- Quizzes on Searching

5. Sorting Algorithm

Sorting algorithms are used to arrange the elements of a list in a specific order, such as numerical or alphabetical. It organizes the items in a systematic way, making it easier to search for and access specific elements.

5. Sorting Algorithm

- Guide on Sorting Algorithms
- Practice problems on Sorting algorithm
- Top Sorting Interview Questions and Problems
- Quiz on Sorting

6. Hash

Hashing is a technique that generates a fixed-size output (hash value) from an input of variable size using mathematical formulas called hash functions. Hashing is commonly used in data structures for efficient searching, insertion and deletion..

6. Hash

- Guide on Hashing
- Practice Problems on Hashing
- Top 20 Hashing Technique based Interview Questions
- Quiz on Hashing

7. Two Pointer Technique

In Two Pointer Technique, we typically use two index variables from two corners of an array. We use the two pointer technique for searching a required point or value in an array.

7. Two Pointer Technique

- Two Pointer Technique
- Practice problems on Two Pointer

8. Window Sliding Technique

In Window Sliding Technique, we use the result of previous subarray to quickly compute the result of current.

8. Window Sliding Technique

- Window Sliding Technique
- Practice problems on Window Sliding

9. Prefix Sum Technique

In Prefix Sum Technique, we compute prefix sums of an array to quickly find results for a subarray.

9. Prefix Sum Technique

- Prefix Sum Technique
- Practice problems on Prefix Sum

10. String

String is a sequence of characters, typically immutable and have limited set of elements (lower case

or all English alphabets).

10. String

- Guide on Strings
- Practice Problems on String
- Top 50 String Coding Problems for Interviews

11. Recursion

Recursion is a programming technique where a function calls itself within its own definition. It is usually used to solve problems that can be broken down into smaller instances of the same problem.

11. Recursion

- Guide on Recursive Algorithms
- Practice Problems on Recursion algorithm
- Top 50 Problems on Recursion Algorithm for Interview
- Quiz on Recursion

12. Matrix/Grid

Matrix is a two-dimensional array of elements, arranged in rows and columns. It is represented as a rectangular grid, with each element at the intersection of a row and column.

12. Matrix/Grid

- Matrix Data Structure Guide
- Practice Problems on Matrix/Grid
- Top 50 Problems on Matrix/Grid for Interviews

13. Stack

Stack is a linear data structure that follows the Last In, First Out (LIFO) principle. Stacks play an important role in managing function calls, memory, and are widely used in algorithms like stock span

problem, next greater element and largest area in a histogram.

13. Stack

- Stack Data Structure Guide
- Practice Problems on Stack
- Top 50 Problems on Stack for Interviews
- Quiz on Stack

14. Queue

Queue is a linear data structure that follows the First In, First Out (FIFO) principle. Queues play an important role in managing tasks or data in order, scheduling and message handling systems.

14. Queue

- Queue Data Structure Guide
- Practice Problems on Queue
- Top 50 Problems on Queue for Interviews
- Quiz on Queue

15. Linked List

Linked list is a linear data structure that stores data in nodes, which are connected by pointers. Unlike arrays, nodes of linked lists are not stored in contiguous memory locations and can only be accessed sequentially, starting from the head of list.

15. Linked List

- Linked List Data Structure Guide
- Practice problems on Linked Lists
- Top 50 Problems on Linked List for Interviews
- Quiz on Linked List

16. Tree

Tree is a non-linear, hierarchical data structure consisting of nodes connected by edges, with a top node called the root and nodes having child nodes. It is widely used in file systems, databases, decision-making algorithms, etc.

16. Tree

- Tree Data Structure Guide
- Practice Problems on Tree
- Top 50 Tree Coding Problems for Interviews
- Quiz on Tree

17. Heap

Heap is a complete binary tree data structure that satisfies the heap property. Heaps are usually used to implement priority queues, where the smallest or largest element is always at the root of the tree.

17. Heap

- Heap Data Structure Guide
- Practice Problems on Heap
- Top 50 Problems on Heap for Interviews
- Quiz on Heap

18. Graph

Graph is a non-linear data structure consisting of a finite set of vertices (or nodes) and a set of edges (or links) that connect a pair of nodes. Graphs are widely used to represent relationships between entities.

18. Graph

- Guide on Graph Algorithms
- Practice Problems on Graph
- Top 50 Problems on Graph for Interviews

- Quiz on Graph

19. Greedy Algorithm

Greedy Algorithm builds up the solution one piece at a time and chooses the next piece which gives the most obvious and immediate benefit i.e., which is the most optimal choice at that moment. So the problems where choosing locally optimal also leads to the global solutions are best fit for Greedy.

19. Greedy Algorithm

- Guide on Greedy Algorithms
- Practice Problems on Greedy Algorithm
- Top 20 Greedy Algorithm Interview Questions
- Quiz on Greedy

20. Dynamic Programming

Dynamic Programming is a method used to solve complex problems by breaking them down into simpler subproblems. By solving each subproblem only once and storing the results, it avoids redundant computations, leading to more efficient solutions for a wide range of problems.

20. Dynamic Programming

- Dynamic Programming Guide
- Practice Problems on Dynamic Programming
- Top 50 Dynamic Programming Coding Problems for Interviews
- Quiz on DP

21. Other Algorithms

Bitwise Algorithms: Operate on individual bits of numbers.

21. Other Algorithms

- Bitwise Algorithms Guide
- Practice Problems on Bit Magic

- Quiz on Bit Magic

21. Other Algorithms

Backtracking Algorithm :Follow Recursionwith the option to revert and traces back if the solution from current point is not feasible.

21. Other Algorithms

- Guide on Backtracking Algorithms
- Practice Problems on Backtracking algorithm
- Top 20 Backtracking Algorithm Interview Questions
- Quiz on Backtracking

21. Other Algorithms

Divide and conquer: A strategy to solve problems by dividing them into smaller subproblems, solving those subproblems, and combining the solutions to obtain the final solution.

21. Other Algorithms

- Guide on Divide and Conquer Algorithm
- Practice problems on Divide And Conquer algorithm
- Quizzes on Divide and Conquer

21. Other Algorithms

Branch and Bound :Used in combinatorial optimization problems to systematically search for the best solution. It works by dividing the problem into smaller subproblems, or branches, and then eliminating certain branches based on bounds on the optimal solution. This process continues until the best solution is found or all branches have been explored.

21. Other Algorithms

- Guide on Branch and Bound Algorithm

21. Other Algorithms

Geometric algorithms are a set of algorithms that solve problems related to shapes, points, lines and polygons.

21. Other Algorithms

- Guide on Geometric Algorithms
- Practice Problem on Geometric Algorithms

21. Other Algorithms

Randomized algorithms are algorithms that use randomness to solve problems. They make use of random input to achieve their goals, often leading to simpler and more efficient solutions. These algorithms may not produce the same result but are particularly useful in situations when a probabilistic approach is acceptable.

21. Other Algorithms

- Guide on Randomized Algorithms

Cheat Sheets

- Blind 75
- SDE SHEET ? A Complete Guide for SDE Preparation
- GeeksforGeeks Master Sheet ? List of all Cheat Sheets

Cheat Sheets

Related Article:

Cheat Sheets

- Introduction to DSA

Cheat Sheets

Master Data Structures and Algorithms at your own pace with our DSA Self-Paced course. In just 90

days, you'll cover core concepts, solve real-world problems, and sharpen your problem-solving skills. Take the 90 Day Challenge: complete 90% of the course in 90 days and get a 90% refund. Stay motivated, track progress, and achieve DSA mastery. Start today!

Cheat Sheets

R

Cheat Sheets

- DSA
- Tutorials
- DSA Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon

- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

HTML List

- [HTML Lists](#)
- [HTML Ordered Lists](#)
- [HTML Unordered Lists](#)
- [HTML Description Lists](#)

HTML Graphics

- [SVG Tutorial](#)

HTML Tutorial References

- [HTML Tags - A to Z List](#)
- [HTML Attributes Complete Reference](#)
- [HTML Global Attributes](#)
- [HTML5 Complete Reference](#)
- [HTML5 MathML Complete Reference](#)
- [HTML DOM Complete Reference](#)
- [HTML DOM Audio/Video Complete Reference](#)
- [SVG Element Complete Reference](#)
- [SVG Attribute Complete Reference](#)
- [SVG Property Complete Reference](#)
- [HTML Canvas Complete Reference](#)

HTML Tutorial

-
-
-

HTML Tutorial

HTML stands for HyperText Markup Language. It is the standard language used to create and

structure content on the web. It tells the web browser how to display text, links, images, and other forms of multimedia on a webpage. HTML sets up the basic structure of a website, and then CSS and JavaScript add style and interactivity to make it look and function better.

HTML Tutorial

How Does HTML Work? - HTML documents are plain-text files saved with an.html extension. Browsers read these documents, interpret the markup (tags and attributes), and render the formatted content on your screen.

HTML Tutorial

To know more about HTML features, history, advantages, etc. - refer to [HTML Introduction](#).

'Hello World' Page in HTML

To show you how fun and easy HTML is, we have provided a classic example of writing "Hello, World!" in HTML

Why Learn HTML?

Learning HTML is essential for various practical reasons:

Why Learn HTML?

- **Foundation of Web Development:** HTML is the starting point for creating websites. Understanding HTML is crucial for any web development or web design role.
- **Universal Language of the Web:** HTML is the standard markup language used to create the structure of web pages. Knowledge of HTML is necessary to manage any content on the Internet.
- **Easy to Learn:** HTML is straightforward compared to programming languages. Beginners can quickly learn how to create basic websites with just HTML.
- **Career Opportunities:** Proficiency in HTML opens up various career paths, including web developer, content manager, and UX/UI designer roles.
- **Gateway to Advanced Technologies:** Once you master HTML, you can easily move on to

learnCSS,JavaScript, and other tools that enhance websites, making them more interactive and visually appealing.

Getting Started with HTML: A Beginner's Guide

ThisHTML tutorialprovides you with a step-by-step learning journey for mastering HTML. You will start with HTML fundamentals, and then move on to advanced HTML5 topics. By the end of this HTML tutorial, you will have a solid foundation in HTML.

1. HTML Basics

By learning the basics of HTML, you can start creating your own web pages and bring your ideas to life online.

1. HTML Basics

- Introduction to HTML
- HTML Editors
- HTML Comments
- HTML Elements
- HTML Attributes
- HTML Doctypes
- HTML Heading
- HTML Paragraphs
- HTML Links
- HTML Images
- HTML Semantics
- HTML Entities
- HTML Symbols

2. HTML Beginner Projects

Start simple and learn by doing beginner HTML projects that teach you how to make basic web

pages, giving you real-world practice in building websites.

2. HTML Beginner Projects

- Simple Portfolio Website
- Design an Event Web Page
- Top 10 Projects For Beginners

3. HTML Basic Tag Questions

HTML tags are the building blocks of web pages. By understanding and using HTML tags effectively, you can create well-structured and informative webpages.

3. HTML Basic Tag Questions

- Add a Paragraph in HTML
- Create a Link in HTML
- Div Tag in HTML
- Span Tag in HTML
- Add a Header on a Webpage
- Add a Footer on a Webpage
- Line Break in HTML
- Add Bavigation Bar in HTML
- Link JavaScript to HTML
- HTML Tags ? A to Z List

4. HTML Tables

HTML tables provide a structured way to organize information into rows and columns, making it easy for users to understand and navigate.

4. HTML Tables

- What is a Table in HTML?

- Add a Table Row in HTML
- Add a Table Header in HTML
- Add a Table Cell in HTML
- Set captions in HTML Table
- Group Columns in HTML

5. HTML Lists

HTML lists organize information clearly on your website. They're perfect for step-by-step instructions, lists of ingredients, or ranking items. This improves readability and helps users to understand your content.

5. HTML Lists

- What are Lists in HTML? Ordered lists in HTML Unordered List in HTML Description Lists in HTML
- Ordered lists in HTML
- Unordered List in HTML
- Description Lists in HTML
- HTML List Elements Add List Items in HTML Create an Unordered list in HTML Create an Ordered list in HTML <dl> Tag in HTML <dt> Tag in HTML <dd> Tag in HTML
- Add List Items in HTML
- Create an Unordered list in HTML
- Create an Ordered list in HTML
- <dl> Tag in HTML
- <dt> Tag in HTML
- <dd> Tag in HTML

5. HTML Lists

- Ordered lists in HTML
- Unordered List in HTML

- Description Lists in HTML

5. HTML Lists

- Add List Items in HTML
- Create an Unordered list in HTML
- Create an Ordered list in HTML
- <dl> Tag in HTML
- <dt> Tag in HTML
- <dd> Tag in HTML

6. HTML Formatting

HTML offers tags and attributes to style your website's text, images, and more. This improves visual appeal, helps users navigate your content, and signals important information to users. Let's explore some of the common HTML formatting tags.

6. HTML Formatting

- Make Text Italic in HTML
- Create Small text in HTML
- Mark Text in HTML
- Add a Subscript in HTML
- Strong Tag in HTML
- Bold Text in HTML
- Highlight Text in HTML
- Show a Deleted Text in HTML
- How to Emphasize Text in HTML
- Add a Superscript in HTML

7. HTML Form

HTML forms let users interact with your website. Use them to collect contact details, run surveys,

search your site, and more. Let's explore more about HTML form:

7. HTML Form

- What is a Form in HTML?
- Add an Input Field in HTML
- Label tag in HTML
- Add a Button in HTML
- Add a Dropdown in HTML
- Add a Textarea in HTML
- Fieldset Tag in HTML
- Legend Tag in HTML
- Datalist Tag in HTML

8. HTML Advanced Concepts

Beyond the Basics in HTML there are also advance concept exists. So, once you have mastered the fundamental building blocks of HTML, we can explore advanced HTML concepts for dynamic, interactive, and meaningful web pages.

8. HTML Advanced Concepts

- iframe in HTML
- File Paths
- Favicon
- Computer Code Elements
- Add Emojis in HTML
- Charsets in HTML
- URL Encoding in HTML
- Responsive Web Design
- HTML Layout

9. HTML Media Elements

- Add Audio to a Webpage
- Add video to a Webpage

10. HTML References

HTML references are essential resources that provide comprehensive information and guidance on the building blocks of webpages. They're perfect for both experienced developers and those just starting out.

10. HTML References

- Tags Reference
- Attributes Reference
- Global Attributes Reference
- Event Attributes Reference
- DOM Reference
- DOM Audio/Videos Reference
- HTML5 Reference

11. HTML Miscellaneous

This HTML tutorial goes beyond the basics! Explore miscellaneous tags and elements to add comments, help search engines, embed multimedia, design forms, use hex color codes, and define specialized content areas.

11. HTML Miscellaneous

- Structure of HTML Document
- Design a web page
- Most commonly used HTML tags
- HTML Form Design
- Design your First Website in Just 1 Week

- Simple Portfolio Website Design
- Design a Portfolio Gallery
- 10 Best HTML Coding Practices You Must Know
- Design a Login Form to an Image using HTML

HTML Tutorial - Prerequisites

This HTML tutorial is perfect for beginners! No prior knowledge is needed, but basic computer skills will help you to get started. After completing this tutorial, you'll have a basic understanding of HTML and be ready for the next stage of web development, CSS.

Best Approach to Learn HTML:

Here's the step-by-step approach to learn and master HTML efficiently and effectively:

Best Approach to Learn HTML:

- Start with Basics: Understand what HTML is and get comfortable with its syntax, including tags, elements, and attributes.
- Practice Basic Tags: Learn to use foundational tags like `<html>`, `<head>`, `<title>`, and `<body>`, and practice formatting text with tags like `<p>`, `<h1>`, and `<a>`.
- Create Simple Web Pages: Apply your skills by building basic web pages. Try making a personal bio or a hobby blog to practice structuring content.
- Learn Intermediate HTML: Get into lists, tables, forms, and semantic elements like `<article>` and `<section>` to organize content better.
- Explore HTML5: Learn advanced features such as multimedia integration using `<audio>` and `<video>`, and understand how to use the `<canvas>` element for drawings.
- Work on Real Projects: Build projects that challenge you, like a full personal portfolio or small business websites to refine your skills and solve real-world problems.
- Stay Updated: Keep learning new HTML features and web standards by following blogs, participating in forums, and practicing with the latest updates.

- Showcase Your Work: Create a professional portfolio to display your projects and share your HTML expertise on platforms like GitHub and LinkedIn.

Best Approach to Learn HTML:

This approach will help you progressively build your HTML skills from fundamental to advanced levels, making you a well-rounded HTML developer and ready for learning CSS and JS.

HTML Online Quiz

Now, To get the mastery in HTML test your knowledge and sharpen your problem-solving skills with HTML quizzes. The below linked quiz can easily enhance your logical skills and problem solving skills.

HTML Online Quiz

- HTML Exercise ? Quiz Set 1
- HTML Exercise ? Quiz Set 2

HTML Examples

Need HTML inspiration? Explore our collection of HTML examples! Find code for hyperlinks, forms, tables, frames, and much more ? all neatly organized for easy learning.

HTML Interview Questions

- HTML Interview Questions for Beginners (2024)
- HTML Intermediate Interview Questions and Answers (2024)
- HTML Interview Questions For Experienced (2024)

HTML CheatSheet

HTML Cheat Sheet is a simple, and quick reference list of basic HTML elements and attributes. The purpose of this Cheat Sheet is to provide you with some quick accurate ready-to-use code snippets and necessary HTML tags and attributes.

HTML Projects

Whether you're a beginner or looking to refine your HTML knowledge, creating projects like a personal portfolio, a basic landing page, or a responsive website can help you understand HTML in a more effective way. Engaging in such projects not only boosts your confidence but also adds value to your web development portfolio.

HTML Projects

- 10 HTML Project Ideas & Topics For Beginners
- Top 10 Projects For Beginners To Practice HTML and CSS Skills

Applications of HTML

HTML (HyperText Markup Language) is versatile in web development, with various practical applications:

Applications of HTML

- Website Development: HTML forms the structure of websites, organizing content like text, images, and links for easy navigation.
- Web Applications: HTML5 enables dynamic forms, multimedia handling, and interactive user experiences directly in browsers.
- Email Templates: HTML helps create visually appealing, responsive email templates that adapt to different devices.
- Game Development: HTML5, paired with JavaScript, allows the creation of simple, lightweight browser games.
- Online Learning & Documentation: HTML structures educational content, making online resources easy to read and navigate.
- Embedding Content: HTML simplifies embedding videos, maps, and social media feeds into web pages.

Careers with HTML in 2024

HTML is the most fundamental technology used for web development. Though expecting a high-paying job with just HTML knowledge can be challenging - if you want to build a successful career, you need to learn additional technologies also. For web development, you should also know CSS, JavaScript, and either a full-stack or front-end framework based on your interests.

Careers with HTML in 2024

That said, there are jobs that require only HTML skills. Here are some examples:

Careers with HTML in 2024

- Web Content Editor: Creates and manages website content using HTML.
- Email Developer: Designs and codes HTML emails for marketing campaigns.
- Landing Page Designer: Builds landing pages to boost marketing conversions.
- SEO Specialist: Optimizes website content using HTML for better search engine ranking.
- HTML Email Tester: Tests HTML emails to ensure they display correctly on different devices.
- Technical Writer: Writes documentation that may need basic HTML knowledge for web publishing.

Careers with HTML in 2024

Geek Tip: While HTML is a great starting point, learning more skills will improve your job prospects and salary

What is HTML?

HTML stands for HyperText Markup Language. It is the standard language used to create and design web pages on the internet. It was introduced by Tim Berners-Lee in 1991 at CERN as a simple markup language. Since then, it has evolved through versions from HTML 2.0 to HTML5 (the latest 2024 version).

What is HTML?

HTML is a combination of Hypertext and Markup language. Hypertext defines the link between the web pages and Markup language defines the text document within the tag.

What does a basic HTML structure look like?

A basic HTML structure consists of the following elements:

Why HTML is important for web development?

HTML, or HyperText Markup Language, is the standard markup language for creating and designing web pages. Understanding HTML is fundamental for anyone entering web development, as it forms the backbone of every web page.

What are HTML tags and how do they work?

HTML uses tags to define elements on a web page. Tags are enclosed in angle brackets ("`<`" "`>`"), and they come in pairs: an opening tag and a closing tag. The content between these tags specifies the element.

What software do I need to learn HTML?

You only need a simple text editor, like Notepad on Windows or TextEdit on Mac. For Output, you'll need a web browser, like Chrome, Firefox, or Safari.

Is HTML difficult to learn for beginners?

HTML is considered one of the easiest programming languages to learn, especially for beginners. Its syntax is very basic, and it provides instant visual feedback, making it an ideal starting point for those new to coding.

What is the difference between HTML and HTML5?

HTML5 is the latest version of HTML, introducing new elements, attributes, and APIs that enhance web development capabilities. Learning HTML5 is recommended for staying current with modern web development practices.

Can I create a complete website with just HTML?

Yes, You can build a simple website with just HTML, but it will be very basic. For a visually appealing and interactive website, you'll also need CSS (for styling) and JavaScript (for dynamic

elements).

Can I create a complete website with just HTML?

Dive deep into the world of JavaScript with our JavaScript Full Course Online, offering everything from the basics to advanced techniques. Learn how to build interactive websites, optimize your code, and earn a certification to showcase your skills. Take the Three 90 Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Push your limits and stay motivated as you work towards mastering JavaScript. Join now and get started on your JavaScript learning journey!

Can I create a complete website with just HTML?

- HTML
- Web Technologies
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems

- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025

- GATE DA Course 2025

Similar Reads

Angular Tutorial

-
-
-

Angular Tutorial

Angular is a powerful, open-source web application framework for building dynamic and scalable single-page applications (SPAs). Developed by Google, Angular provides a comprehensive solution for front-end development with tools for routing, form handling, HTTP services, and more.

Angular Tutorial

- Designed for building scalable web applications
- Component-based architecture for efficient development
- Supports powerful features like two-way data binding, routing, and dependency injection
- Ideal for single-page applications (SPAs) and progressive web apps (PWAs)

Angular Tutorial

To begin developing with Angular, you first need to set up your development environment by installing Node.js and Angular CLI.

Angular Tutorial

- Download and Install Node.js and NPM
- Angular CLI | Angular Project Setup
- Local Environment Setup in Angular
- Getting Started with Angular

Angular Tutorial

Let us now take a look at our first code example.

Angular Tutorial

Now, open your browser and visit <http://localhost:4200> to see your Angular app in action.

Angular Tutorial

In this example

Angular Tutorial

- The AppComponent class is decorated with @Component, which defines its selector, template, and styles.
- The template uses Angular's interpolation syntax to display a simple "Welcome to the Angular Tutorial" message.
- The AppComponent class defines the component logic and properties.

Why Learn Angular?

- Simplifies Front-End Development:Angular provides all the tools needed for front-end development in a single framework, making it easier to build dynamic and scalable applications.
- Component-Based Architecture:Angular's component-based structure helps break down large applications into smaller, manageable parts, making development and maintenance easier.
- Powerful Features:Angular offers powerful features such as two-way data binding, routing, and form validation that speed up the development process.
- Seamless Integration:Angular integrates well with RESTful APIs, making it ideal for building full-stack web applications and SPAs.
- Strong Community Support:With strong backing from Google and an active community, Angular is continuously updated and improved, ensuring long-term stability.

Why Learn Angular?

Angular tutorial Prerequisites:HTML,JavaScript,TypeScript

Angular Basics

- Introduction to Angular

- Setting up Angular Development Environment
- Creating Your First Angular App
- Architecture of Angular Applications
- Components and Templates
- Data Binding in Angular
- Directives in Angular
- Services and Dependency Injection
- Angular Modules
- Pipes in Angular

Angular Versions

- Angular 2
- Angular 4
- Angular 5
- Angular 6
- Angular 7
- Angular 8
- Angular 9
- Angular 10
- Angular 11
- Angular 12
- Angular 13
- Angular 14
- Angular 15
- Angular 16
- Angular 17
- Angular 18

Angular Projects

- ToDo List
- Simple Web-app
- Weather App
- Expense Tracker
- Portfolio Website
- Dice Rolling App
- Movie App
- Color Picker App
- Quiz App
- Joke Generator App
- BMI Calculator
- Snake Game
- Event Calender
- Online Gift Store

Angular For Interview

- Angular Interview Questions and Answers
- Angular Exercises, Practice Questions and Solutions
- AngularJS Quiz | Set-1
- AngularJS Quiz | Set-2
- AngularJS Quiz | Set-3

Features of Angular

- It uses components and directives. Components are the directives with a template.
- It is written in Microsoft's TypeScript language, which is a superset of ECMAScript 6 (ES6).
- Angular is supported by all the popular mobile browsers.
- Properties enclosed in `{}?` and `[]?` are used to bind data between the view and the model.

- It provides support for TypeScript and JavaScript.
- Angular uses `@Route Config{()}` for routing configuration.
- It has a better structure compared to AngularJS, easier to create and maintain for large applications but behind AngularJS in the case of small applications.
- It comes with the Angular CLI tool.

Applications of Angular

- Single-Page Applications (SPAs): Angular excels at creating SPAs, where a single webpage dynamically updates content as users interact with it.
- Real-Time Applications: Use Angular for chat applications, weather forecasts, and live tracking systems that require immediate updates.
- Enterprise Applications: Angular is commonly used for developing large-scale enterprise applications. Its modular architecture, dependency injection, and TypeScript support make it suitable for building complex and maintainable applications.
- Content Management Systems (CMS): Angular can be used to build custom content management systems, providing a dynamic and responsive user interface for managing content.

History of Angular

- Angular version 1.0 was released in 2012.
- Misko Hevery, an employee at Google, started to work with AngularJS in 2009.
- The project is now officially supported by Google, and the idea turned out very well.

Angular CheatSheet

The cheat sheet is a quick reading manual that will help you to revise all the concepts at a glance. The detailed description is covered in the [Angular Cheat Sheet ? A Basic Guide to Angular](#) Article.

What is Angular?

Angular is a TypeScript-based open-source framework for building client-side applications. It is developed and maintained by Google.

How does Angular works?

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript.

What are the key features of Angular?

Key features of angular are Angular Directives, Two way data binding, Angular CLI and Routing.

What are Angular Directives?

The Directive is generally a built-in class that includes the additional behavior to elements in the Angular Applications.

What are Decorators in Angular?

The Decorators are the function that is called with the prefix @ symbol, immediately followed by the class, methods, or property.

What is Angular CLI?

Angular has its own command-line interface that provides simple and easy steps to create, develop, build, and deploy Angular Applications.

What are Angular Pipes?

Pipes in Angular can be used to transform the strings, currency amounts, dates, and other data, without affecting the actual content.

What are Angular Pipes?

Become a Full-Stack Developer and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

What are Angular Pipes?

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you

need?Start the challenge right away!

What are Angular Pipes?

J

What are Angular Pipes?

- Web Technologies
- AngularJS
- Web-Tech Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)

- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects

- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design

- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Aptitude for Placements

-
-
-

Aptitude for Placements

An aptitude test is quite vital for placements as it helps in enhancing the logical skills of the person and if someone is preparing for interviews then aptitude preparation is very much required. Hiring can be both off-campus and on-campus but the procedure of selection is almost the same that is sorting out candidates on the basis of basic aptitude tests. If going for programming placement maths is a must, as it is the most important aspect of coding.

Aptitude for Placements

Aptitude for Placements

Aptitude for Placements

The recruiters ask questions regarding logical reasoning, quantitative aptitude, data interpretation, and verbal ability. So if you are going for placements then you must prepare well and these things are also useful for competitive examinations, entrance tests, interviews, government examinations, etc. The aptitude test is conducted offline or online depending on the companies. Mostly the mass recruiters companies such as TCS, Wipro, Capgemini, Cognizant, and Infosys conduct their screening test i.e. the aptitude test online as it is quite easy to evaluate.

Aptitude for Placements

This selection process mostly includes-

Aptitude for Placements

- Aptitude Test
- Coding Round/Group Discussion

- Multiple Face To Face Interviews (including the HR Round)

What is Aptitude Test?

An aptitude test or exam is something that is used for determining the skills of an individual. With the help of an aptitude test, one can assume the strength and weaknesses of an individual and their inclination towards success and failure.

What is Aptitude Test?

The four basic types of aptitude tests are logical aptitude, spatial aptitude, organizational aptitude, and physical aptitude. Also, aptitude is used in companies for testing the mental awareness, problem-solving skills, and accuracy of the candidate.

Why are Aptitude Tests Important for Campus Placement?

An aptitude test is important for campus placement as it helps in evaluating the employability skills of the candidate. Good aptitude skills allow the person to stand differently in the crowd. Aptitude tests also help in finding suitable graduates by making a precise assessment based on logical, analytical, verbal, and numerical abilities.

Quantitative Aptitude

- Numbers (Learn|Practice)
- LCM and HCF (Learn|Practice LCM|Practice HCF)
- Work and Wages (Learn|Practice)
- Pipes and Cisterns (Learn|Practice)
- Time Speed Distance (Learn|Practice)
- Trains, Boats, and Streams (Learn|Practice)
- Percentages (Learn|Practice)
- Ratio Proportion and Partnership (Learn|Practice)
- Mixtures and Alligation (Learn|Practice)
- Algebra (Learn|Practice)

- Average (Learn|Practice)
- Age (Learn|Practice)
- Profit and Loss (Learn|Practice)
- Simple Interest (Learn|Practice)
- Compound Interest (Learn|Practice)
- Mensuration 2D (Learn|Practice)
- Mensuration 3D (Learn|Practice)
- Trigonometry & Height and Distances (Learn|Practice)
- Progressions (Learn|Practice)
- Logarithms (Learn| Practice)
- Permutation and Combination (Learn|Practice)
- Probability (Learn|Practice)
- Geometry (Learn|Practice)
- Clocks (Learn|Practice)
- Calendars (Learn|Practice)
- Coding-Decoding (Learn|Practice)
- Race (Learn|Practice)
- Simplification and Approximation (Learn |Practice)

Verbal Ability

- Passage/Sentence Rearrangement (Learn)
- Error Detection and Correction (Learn)
- Fill in the Blanks (Learn)
- Comprehension Passages (Learn)
- Cloze test (Learn)
- Jumbled words/Paragraphs (Learn)
- Antonyms and synonyms (Learn)

- Idioms and phrases (Learn)
- Vocabulary and grammar (Learn)
- One word Substitution (Learn)

Logical Reasoning

- Series: Missing Numbers, Odd One Out(Learn|Practice)
- Data Sufficiency (Learn|Practice)
- Assumptions and Conclusions, Courses of Action (Learn|Practice)
- Puzzles (Learn|Practice)
- Syllogism (Learn|Practice)
- Cubes (Placement | Cubes)
- Alphanumeric series (Learn)
- Reasoning Analogy (Learn)
- Blood Relations (Learn)
- Calendars (Learn)
- Clocks (Learn)
- Dices (Learn)
- Deductive Reasoning/Statement Analysis (Learn)

Q1. What are the aptitude topics for placement preparation?

Answer:

Q1. What are the aptitude topics for placement preparation?

These are a few aptitude topics for placement preparation:

Q1. What are the aptitude topics for placement preparation?

- Verbal ability
- Data sufficiency and interpretation
- Logical reasoning

- Numerical ability and Quantitative aptitude

Q2. How can I practice for the aptitude test?

Answer:

Q2. How can I practice for the aptitude test?

These are the ways in which one can prepare for an aptitude test:

Q2. How can I practice for the aptitude test?

- Practice properly
- Know your strengths
- Research deeply
- Manage time appropriately
- Stay focused

Q3. What is the trick to solving aptitude questions?

Answer:

Q3. What is the trick to solving aptitude questions?

Here are the tricks to solve aptitude questions:

Q3. What is the trick to solving aptitude questions?

- Focus on your basics
- Take care of your speed
- Stay focused on weaker links
- Solve rare problems
- Strengthen your weaker links

Q4. How can I increase my aptitude speed?

Answer:

Q4. How can I increase my aptitude speed?

These are the ways to increase aptitude speed:

Q4. How can I increase my aptitude speed?

- Understand the question
- Try to learn the speed and Vedic maths
- Focus on your weaker areas
- Go through the complete syllabus thoroughly

Q4. How can I increase my aptitude speed?

Dreaming of M.Tech in IIT? Get AIR under 100 with our GATE 2026 CSE & DA courses! Get flexible weekday/weekend options, live mentorship, and mock tests. Access exclusive features like All India Mock Tests, and Doubt Solving? your GATE success starts now!

Q4. How can I increase my aptitude speed?

- Computer Subject
- Career-Advices
- placement
- placement preparation

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media

- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS

- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git

- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter

- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course

- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification

- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

C Basics

- C Language Introduction
- Features of C Programming Language
- C Programming Language Standard
- C Hello World Program
- Compiling a C Program: Behind the Scenes
- C Comments
- Tokens in C
- Keywords in C

C Variables and Constants

- C Variables
- Constants in C
- Const Qualifier in C
- Different ways to declare variable as constant in C
- Scope rules in C
- Internal Linkage and External Linkage in C
- Global Variables in C

C Data Types

- Data Types in C
- Literals in C
- Escape Sequence in C
- bool in C
- Integer Promotions in C
- Character Arithmetic in C
- Type Conversion in C

C Input/Output

- Basic Input and Output in C
- Format Specifiers in C
- printf in C
- scanf in C
- Scansets in C
- Formatted and Unformatted Input/Output functions in C with Examples

C Operators

- Operators in C
- Arithmetic Operators in C
- Unary operators in C
- Relational Operators in C
- Bitwise Operators in C
- C Logical Operators
- Assignment Operators in C
- Increment and Decrement Operators in C
- Conditional or Ternary Operator (?:) in C
- sizeof operator in C
- Operator Precedence and Associativity in C

C Control Statements Decision-Making

- Decision Making in C (if , if..else, Nested if, if-else-if)
- C - if Statement
- C if else Statement
- C if else if ladder
- Switch Statement in C
- Using Range in switch Case in C

- C - Loops
- C for Loop
- while Loop in C
- do...while Loop in C
- For vs. While
- Continue Statement in C
- Break Statement in C
- goto Statement in C

C Functions

- C Functions
- User-Defined Function in C
- Parameter Passing Techniques in C
- Function Prototype in C
- How can I return multiple values from a function?
- main Function in C
- Implicit Return Type int in C
- Callbacks in C
- Nested functions in C
- Variadic Functions in C
- `_Noreturn` function specifier in C
- Predefined Identifier `__func__` in C
- C Library `math.h` Functions

C Arrays & Strings

- C Arrays
- Properties of Array in C
- Multidimensional Arrays in C - 2D and 3D Arrays

- Initialization of Multidimensional Array in C
- Pass Array to Functions in C
- How to pass a 2D array as a parameter in C?
- What are the data types for which it is not possible to create an array?
- How to pass an array by value in C ?
- Strings in C
- Array of Strings in C
- What is the difference between single quoted and double quoted declaration of char array?
- C String Functions

C Pointers

- C Pointers
- Pointer Arithmetics in C with Examples
- C - Pointer to Pointer (Double Pointer)
- Function Pointer in C
- How to Declare a Pointer to a Function?
- Pointer to an Array | Array Pointer
- Difference between constant pointer, pointers to constant, and constant pointers to constants
- Pointer vs Array in C
- Dangling, Void , Null and Wild Pointers in C
- Near, Far and Huge Pointers in C
- restrict Keyword in C

C User-Defined Data Types

- C Structures
- dot (.) Operator in C
- C typedef
- Structure Member Alignment, Padding and Data Packing

- Flexible Array Members in a structure in C
- C Unions
- Bit Fields in C
- Difference Between Structure and Union in C
- Anonymous Union and Structure in C
- Enumeration (or enum) in C

C Storage Classes

- Storage Classes in C
- extern Keyword in C
- Static Variables in C
- Initialization of Static Variables in C
- Static Functions in C
- Understanding "volatile" qualifier in C | Set 2 (Examples)
- Understanding "register" keyword in C

C Memory Management

- Memory Layout of C Programs
- Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()
- Difference Between malloc() and calloc() with Examples
- What is Memory Leak? How can we avoid?
- Dynamic Array in C
- How to dynamically allocate a 2D array in C?
- Dynamically Growing Array in C

C Preprocessor

- C Preprocessors
- C Preprocessor Directives

- How a Preprocessor works in C?
- Header Files in C
- What's difference between header files "stdio.h" and "stdlib.h" ?
- How to write your own header file in C?
- Macros and its types in C
- Interesting Facts about Macros and Preprocessors in C
- # and ## Operators in C
- How to print a variable name in C?
- Multiline macros in C
- Variable length arguments for Macros
- Branch prediction macros in GCC
- Difference between typedef and #define in C
- Difference between #define and const in C

C File Handling

- Basics of File Handling in C
- C fopen() Function
- EOF, getc() and feof() in C
- fgets() in C
- fseek() vs rewind() in C
- What is return type of getchar(), fgetc() and getc() ?
- Read/Write Structure From/to a File in C
- C Program to Print Contents of File
- C Program to Delete a File
- C Program to Merge Contents of Two Files into a Third File
- What is the difference between printf, sprintf and fprintf?
- Difference between getc(), getchar(), getch() and getche()

Miscellaneous

- time.h header file in C with Examples
- Input-output system calls in C | Create, Open, Close, Read, Write
- Signals in C language
- Program error signals
- Socket Programming in C
- _Generics Keyword in C
- Multithreading in C

C Interview Questions

- C Programming Interview Questions (2025)
- Commonly Asked C Programming Interview Questions | Set 1
- Commonly Asked C Programming Interview Questions | Set 2
- Commonly Asked C Programming Interview Questions | Set 3

C Programming Language Tutorial

-
-
-

C Programming Language Tutorial

C is a general-purpose, procedural, and middle-level programming language used for developing computer software, system programming, applications, games, and more. Known for its simplicity and efficiency, C is an excellent choice for beginners as it provides a strong foundation in programming concepts. C was developed by Dennis M. Ritchie at Bell Laboratories in 1972. Initially, it was created for programming the UNIX operating system.

C Programming Language Tutorial

- Bridges the gap between low-level programming (closer to hardware and machine

code) and high-level programming (closer to human-readable code).

- Referred as the "mother of all programming languages" because it influenced many modern programming languages like C++, Java, Python and Go.
- C programs are platform-independent i.e. code can be compiled and run on different systems with minimal modifications.
- Does not require heavy runtime environments or libraries, making it ideal for low-resource systems.

Hello World Program

In this C tutorial, we'll cover everything from basic syntax, data types, and control structures to advanced topics like pointers, memory management, and file handling. By the end, you'll gain hands-on experience and a solid understanding of C, which is essential for mastering other programming languages like C++ and Java. Let's dive into the world of C programming and build a strong coding foundation!

C Overview

- What is C Language?
- C Features
- C Standards
- Setting Up C Development Environment
- Hello World Program
- Compiling a C Program: Behind the Scenes

C Basics

- Tokens
- Identifiers
- Keywords
- Comments
- Variables

- Constants
- Data Types
- Data Type Modifiers
- Type Conversion
- Operators

C Input/Output

- Basic Input and Output
- printf
- scanf
- Format Specifiers
- Escape Sequence

C Operators

- Arithmetic Operators
- Unary Operators
- Assignment Operators
- Logical Operators
- Bitwise Operator
- Signed Number Representation
- Operator Precedence and Associativity

C Flow Control

- if Statement
- if?else Statement
- if-else-if Ladder
- Switch Statement
- Using Range in Switch Case

- for Loop
- while looping
- do?while Loop
- continue Statement
- break Statement
- goto Statement

C Functions

- Introduction to Functions
- User-Defined Function
- Parameter Passing Techniques
- Main Function
- Inline Function
- Nested Functions

C Arrays

- Introduction to Arrays
- Length of Array
- Multidimensional Arrays
- Pass Array to Functions
- Pass a 2D Array as a Parameter

C Strings

- Introduction to Strings
- Length of String
- String Comparison
- String Copy
- String Concatenation

- Array of Strings
- String Functions

C Pointers

- Introduction to Pointers
- Pointer Arithmetics
- Pointer to Pointer (Double Pointer)
- Function Pointer
- Declare Function Pointer
- Pointer to an Array
- Constant Pointer
- Pointer vs Array
- Dangling, Void, Null and Wild Pointers
- Near, Far and Huge Pointers
- restrict Keyword

C Dynamic Memory Allocation

- Memory Structure of a Program
- Dynamic Memory Allocation (malloc(), calloc(), and free())
- Memory Leak

C Structures and Union

- Introduction to Structures
- Array of Structures
- Structure Pointer
- Structure Member Alignment
- Introduction to Unions
- Bit Fields

- Structure vs Union
- Anonymous Union and Structure
- Enumeration (or enum)
- dot (.) Operator
- typedef

C Storage Classes

- Introduction to Storage Classes
- extern Keyword
- Static Variables
- Initialization of Static Variables
- Static Functions
- Understanding ?volatile? Qualifier
- Understanding the ?register? Keyword

C Preprocessor

- Introduction to Preprocessors
- Preprocessor Directives
- How a Preprocessor Works?
- Header Files
- Header Files ?stdio.h? vs ?stdlib.h?
- Write Your Own Header File
- Macros and its Types
- Interesting Facts About Macros and Preprocessors
- # and ## Operators
- Print a Variable Name
- Multiline Macros
- Variable Length Arguments for Macros

- Branch Prediction Macros in GCC
- typedef versus #define
- Difference Between #define and const

C File Handling

- Basics of File Handling
- fopen() Function
- EOF, getc() and feof()
- fgets() and gets()
- fseek() vs rewind()
- Return Type of getchar(), fgetc() and getc()
- Read/Write Structure From/to a File
- Print Contents of File
- Delete a File
- Merge Contents of Two Files into a Third File
- printf vs sprintf vs fprintf
- getc() vs getchar() vs getch() vs getche()

C Error Handling

- Error Handling
- Using goto for Exception Handling
- Error Handling During File Operations
- Handle Divide By Zero and Multiple Exceptions

C Programs

- Basic C Programs
- Control Flow Programs
- Pattern Printing Programs

- Functions Programs
- Arrays Programs
- Strings Programs
- Conversions Programs
- Pointers Programs
- Structures and Unions Programs
- File I/O Programs
- Date and Time Programs
- More C Programs

Miscellaneous

- Date and Time
- Input-Output System Calls
- Signals
- Program Error Signals
- Socket Programming
- _Generics Keyword
- Multithreading

C Interview Questions

- Top 50 C Programming Interview Questions and Answers
- Commonly Asked C Programming Interview Questions | Set 1
- Commonly Asked C Programming Interview Questions | Set 2
- Commonly Asked C Programming Interview Questions | Set 3

C Compiler

C compiler is a software that translates human-readable C language code into machine code or an intermediate code that can be executed by a computer's central processing unit (CPU).

C Compiler

There are many C compilers available in the market, such as GNU Compiler Collection (GCC), Microsoft Visual C++ Compiler, Clang, Intel C++ Compiler, and TinyCC (TCC).

C Compiler

For this tutorial, we will be using the GNU-based online C compiler provided by GeeksforGeeks which is developed for beginners and is very easy to use compared to other compiler/IDE's available on the web.

Features of C Language

There are some key features of C language that show the ability and power of C language:

Features of C Language

- **Simplicity and Efficiency:** The simple syntax and structured approach make the C language easy to learn.
- **Fast Speed:** C is one of the fastest programming language because C is a static programming language, which is faster than dynamic languages like Javascript and Python. C is also a compiler-based which is the reason for faster code compilation and execution.
- **Portable:** C provides the feature that you write code once and run it anywhere on any computer. It shows the machine-independent nature of the C language.
- **Memory Management:** C provides lower level memory management using pointers and functions like `realloc()`, `free()`, etc.
- **Pointers:** C comes with pointers. Through pointers, we can directly access or interact with the memory. We can initialize a pointer as an array, variables, etc.
- **Structured Language:** C provides the features of structural programming that allows you to code into different parts using functions which can be stored as libraries for reusability.

Applications of C Language

C was used in programs that were used in making operating systems. C was known as a system

development language because the code written in C runs as fast as the code written in assembly language.

Applications of C Language

The use of C is given below:

Applications of C Language

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

1. How to learn C easily?

The first steps towards learning C or any language are to write a hello world program. It gives the understanding of how to write and execute a code. After this, learn the following:

1. How to learn C easily?

- Variables
- Operators
- Conditionals
- Loops and Errors
- Arrays and Strings
- Pointers and Memory

- Functions
- Structures
- Recursions

2. Difference between C and C++?

C | CPP

C is a procedural programming language. | C++ is both a procedural and object-oriented programming language.

It does not support Function overloading. | It supports function overloading.

Operator overloading is not supported. | Operator overloading is supported.

C does not support data hiding which leads to security concerns. | Data hiding is supported in C++ by Data Encapsulation.

2. Difference between C and C++?

C

2. Difference between C and C++?

CPP

2. Difference between C and C++?

C is a procedural programming language.

2. Difference between C and C++?

C++ is both a procedural and object-oriented programming language.

2. Difference between C and C++?

It does not support Function overloading.

2. Difference between C and C++?

It supports function overloading.

2. Difference between C and C++?

Operator overloading is not supported.

2. Difference between C and C++?

Operator overloading is supported.

2. Difference between C and C++?

C does not support data hiding which leads to security concerns.

2. Difference between C and C++?

Data hiding is supported in C++ by Data Encapsulation.

3. Is C easy to learn for beginners?

While C is one of the easy languages, it is still a good first language choice to start with because almost all programming languages are implemented in it. It means that once you learn C language, it'll be easy to learn more languages like C++, Java, and C#.

4. Why should we learn C first rather than C++?

C is a 'mother of all languages.' It provides a solid understanding of fundamental programming concepts and is considered easier to grasp. C offers versatile applications, from software development to game programming, making it an excellent choice for building a strong programming foundation.

4. Why should we learn C first rather than C++?

Master C programming with our C Programming Course Online, which covers everything from the basics to advanced concepts like data structures. Gain a deep understanding of C and enhance your problem-solving abilities with practical coding challenges. Take the Three 90 Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Track your learning journey and get rewarded for your commitment. Join now and start mastering C programming!

4. Why should we learn C first rather than C++?

- C Language
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation

- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp

- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux

- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures

- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

CI/CD Pipeline - System Design

-
-
-

CI/CD Pipeline - System Design

CI/CD pipeline automates software development, making code integration, testing, and deployment faster and more reliable. In CI/CD Pipeline -System Design, we explore the stages and tools involved in creating a smooth workflow that helps teams deliver high-quality software quickly. We'll break down how continuous integration (CI) combines code changes into a shared repository, while continuous delivery (CD) ensures that code is always ready to deploy.

CI/CD Pipeline - System Design

Important Topics for CI/CD Pipeline

CI/CD Pipeline - System Design

- What is CI/CD?
- Importance of CI/CD in Modern Software Development
- Components of a CI/CD Pipeline
- Steps for Designing a CI/CD Pipeline
- Setting Up a CI/CD Pipeline
- Pipeline Orchestration
- Security in CI/CD Pipelines
- Scaling CI/CD Pipelines
- Common Challenges with CI/CD Pipeline and How to Avoid Them

What is CI/CD?

CI/CD stands for Continuous Integration and Continuous Deployment (or Continuous Delivery) and is an essential practice in modern software development. It focuses on automating and streamlining

the process of integrating code changes, testing, and deploying software. Here's a brief overview of each component in the context of system design:

Continuous Integration (CI)

- **Integration of Code:** Developers frequently merge their code changes into a shared repository, usually several times a day. This practice helps in detecting integration issues early.
- **Automated Testing:** Each integration triggers an automated build and testing sequence to ensure that the new code changes do not break the existing functionality.
- **Feedback:** Quick feedback is provided to developers on the quality and correctness of their code, allowing for rapid identification and correction of issues.

Continuous Deployment (CD)

- **Continuous Delivery:** This ensures that the codebase is always in a deployable state. While not every change is automatically deployed to production, the system is designed to support such deployments at any time.
- **Continuous Deployment:** Extends continuous delivery by automatically deploying every change that passes automated tests to production without human intervention.

Importance of CI/CD in Modern Software Development

Continuous Integration and Continuous Deployment (CI/CD) are vital in modern software development for several reasons:

Importance of CI/CD in Modern Software Development

- **Faster Time to Market:** Quick Iterations: CI/CD allows for rapid iteration of software, enabling new features and updates to reach users faster.
- **Automated Processes:** Automation reduces the time needed for manual testing and deployment, speeding up the development cycle.
- **Quick Iterations:** CI/CD allows for rapid iteration of software, enabling new features and updates to reach users faster.
- **Automated Processes:** Automation reduces the time needed for manual testing and deployment,

speeding up the development cycle.

- Improved Code Quality:Automated Testing:Continuous testing ensures that new code changes do not introduce bugs or break existing functionality.Consistent Integration:Regular integration of code helps in identifying and resolving conflicts early.

- Automated Testing:Continuous testing ensures that new code changes do not introduce bugs or break existing functionality.

- Consistent Integration:Regular integration of code helps in identifying and resolving conflicts early.

- Increased Collaboration and Efficiency:Shared Codebase:Developers work on a single shared codebase, improving collaboration and reducing integration issues.Feedback Loops:Automated feedback on code changes helps developers fix issues quickly, enhancing overall productivity.

- Shared Codebase:Developers work on a single shared codebase, improving collaboration and reducing integration issues.

- Feedback Loops:Automated feedback on code changes helps developers fix issues quickly, enhancing overall productivity.

- Reduced Risks:Early Bug Detection:Frequent integration and testing catch bugs early in the development cycle, reducing the risk of critical failures in production.Rollback Capability:Automated deployments often include easy rollback options, minimizing the impact of any issues that do make it to production.

- Early Bug Detection:Frequent integration and testing catch bugs early in the development cycle, reducing the risk of critical failures in production.

- Rollback Capability:Automated deployments often include easy rollback options, minimizing the impact of any issues that do make it to production.

- Consistent and Reliable Releases:Repeatable Processes:Automation ensures that the process of building, testing, and deploying software is consistent every time.Reduced Human Error:Automation minimizes manual intervention, reducing the likelihood of errors.

- Repeatable Processes:Automation ensures that the process of building, testing, and deploying software is consistent every time.

- **Reduced Human Error:**Automation minimizes manual intervention, reducing the likelihood of errors.
- **Scalability and Flexibility:**Handling Load:CI/CD pipelines can handle multiple builds and deployments simultaneously, making it easier to scale development efforts.Adapting to Changes:The flexibility of CI/CD pipelines allows teams to adapt quickly to changes in requirements or technology.
- **Handling Load:**CI/CD pipelines can handle multiple builds and deployments simultaneously, making it easier to scale development efforts.
- **Adapting to Changes:**The flexibility of CI/CD pipelines allows teams to adapt quickly to changes in requirements or technology.

Importance of CI/CD in Modern Software Development

- **Quick Iterations:**CI/CD allows for rapid iteration of software, enabling new features and updates to reach users faster.
- **Automated Processes:**Automation reduces the time needed for manual testing and deployment, speeding up the development cycle.

Importance of CI/CD in Modern Software Development

- **Automated Testing:**Continuous testing ensures that new code changes do not introduce bugs or break existing functionality.
- **Consistent Integration:**Regular integration of code helps in identifying and resolving conflicts early.

Importance of CI/CD in Modern Software Development

- **Shared Codebase:**Developers work on a single shared codebase, improving collaboration and reducing integration issues.
- **Feedback Loops:**Automated feedback on code changes helps developers fix issues quickly, enhancing overall productivity.

Importance of CI/CD in Modern Software Development

- Early Bug Detection:Frequent integration and testing catch bugs early in the development cycle, reducing the risk of critical failures in production.
- Rollback Capability:Automated deployments often include easy rollback options, minimizing the impact of any issues that do make it to production.

Importance of CI/CD in Modern Software Development

- Repeatable Processes:Automation ensures that the process of building, testing, and deploying software is consistent every time.
- Reduced Human Error:Automation minimizes manual intervention, reducing the likelihood of errors.

Importance of CI/CD in Modern Software Development

- Handling Load:CI/CD pipelines can handle multiple builds and deployments simultaneously, making it easier to scale development efforts.
- Adapting to Changes:The flexibility of CI/CD pipelines allows teams to adapt quickly to changes in requirements or technology.

1. Commit Change

Developers make code changes and commit these changes to a version control system (e.g., Git). This step initiates the CI/CD pipeline. Committing code changes ensures that they are tracked and versioned properly.

2. Trigger Build

The version control system detects the new commit and triggers the build process automatically. Automated triggering of the build process ensures that new changes are continuously integrated and tested.

3. Build

The codebase is compiled and built into a deployable artifact, such as a binary executable or a

Docker image. Common tools include Maven, Gradle, Ant for Java projects, and Docker for containerized applications. The build step verifies that the code compiles correctly and that all dependencies are resolved.

4. Notify of Build Outcome

The CI/CD system notifies the team of the build results, whether it passed or failed. Immediate feedback on the build status helps developers quickly identify and resolve any build issues. Notification systems can include emails, chat integrations (like Slack or Microsoft Teams), or dashboards in tools like Jenkins or GitLab CI.

5. Run Tests

Automated tests are executed on the build artifact. These can include unit tests, integration tests, end-to-end tests, and more. Testing frameworks like JUnit, Selenium, TestNG, pytest, etc. Running tests ensures that the new code does not introduce any bugs or regressions and that it meets the required quality standards.

6. Notify of Test Outcome

The results of the test suite are reported back to the development team. Quick feedback on test outcomes allows developers to address any failures promptly. Similar to build notifications, results can be sent via email, chat, or dashboards.

7. Deliver Build to Staging

If tests pass, the build artifact is deployed to a staging environment. The staging environment simulates the production environment, allowing for final validation before production deployment. Deployment tools like Ansible, Chef, Puppet, Kubernetes, or cloud-specific services like AWS CodeDeploy.

8. Deploy to Production

After successful validation in staging, the build is automatically or manually promoted to the

production environment. This step makes the new features and fixes available to end-users. Deployment strategies might include blue-green deployment, canary releases, or rolling updates to minimize downtime and risk.

Steps for Designing a CI/CD Pipeline

Designing a CI/CD pipeline involves several key steps to ensure a smooth and efficient workflow from code commit to deployment. Here are the steps to design an effective CI/CD pipeline:

Step 1: Assess Current Development Process

- Understand Current Workflow: Map out your existing development, testing, and deployment processes.
- Identify Bottlenecks: Identify areas where automation can save time or reduce errors.
- Set Objectives: Define the goals of your CI/CD pipeline (e.g., faster releases, improved quality, reduced manual intervention).

Step 2: Choose Tools and Technologies

- Version Control System (VCS): Select a VCS like Git, GitHub, GitLab, or Bitbucket.
- CI/CD Tools: Choose CI/CD tools that fit your needs. Common options include Jenkins, GitLab CI, CircleCI, Travis CI, and Azure DevOps.
- Build Tools: Select build tools appropriate for your technology stack (e.g., Maven, Gradle, npm).
- Testing Frameworks: Choose frameworks for automated testing (e.g., JUnit, Selenium, pytest).

Step 3: Define Pipeline Stages

- Commit Stage: Ensure code changes trigger the pipeline automatically.
- Build Stage: Configure the build process to compile code and package artifacts.
- Test Stage: Automate running of unit tests, integration tests, and other relevant tests.
- Staging Deployment: Deploy to a staging environment for further validation.
- Production Deployment: Define the process for deploying to the production environment.

Step 4: Implement Automated Testing

- Unit Tests: Ensure every code change passes a suite of unit tests.
- Integration Tests: Validate the interaction between different modules.
- End-to-End Tests: Test the application from a user perspective.
- Performance Tests: Ensure the application meets performance requirements.

Step 5: Set Up Continuous Integration

- Automate Builds: Set up the CI tool to automatically build the code upon each commit.
- Run Tests: Integrate automated tests into the CI process to run with every build.
- Provide Feedback: Ensure the CI tool provides immediate feedback to developers on build and test results.

Step 6: Implement Continuous Delivery

- Automate Deployments to Staging: Configure automated deployment to a staging environment after successful builds and tests.
- Manual or Automated Approvals: Define whether deployments to production require manual approval or are automated.

Step 7: Implement Continuous Deployment (Optional)

- Automate Production Deployments: If desired, automate the deployment process to production, ensuring all tests pass and conditions are met.

Step 8: Monitor and Improve

- Logging and Monitoring: Set up logging and monitoring to track the health and performance of the pipeline and deployed applications.
- Feedback Loops: Collect feedback from developers and stakeholders to continuously improve the pipeline.
- Iterate and Optimize: Regularly review and optimize the CI/CD process for efficiency and effectiveness.

Step 9: Security and Compliance

- Security Scans: Integrate security checks into the pipeline to identify vulnerabilities.
- Compliance Checks: Ensure the pipeline enforces compliance with industry standards and regulations.

Step 10: Documentation and Training

- Document Processes: Create clear documentation for the pipeline process, including setup, usage, and troubleshooting.
- Training: Provide training for the development team on how to use the CI/CD pipeline effectively.

Setting Up a CI/CD Pipeline

Setting up a CI/CD pipeline involves several stages, from configuring your version control system to deploying your application. Below are detailed steps to set up a CI/CD pipeline using a common tool like Jenkins, but the concepts can be applied to other CI/CD tools as well:

Step 1. Install and Configure Jenkins

- Download Jenkins: Go to the Jenkins website and download the appropriate installer for your operating system.
- Install Jenkins: Follow the installation instructions for your operating system.
- Start Jenkins: After installation, start Jenkins and access it via <http://localhost:8080>.

Step 2. Set Up Version Control System (VCS)

- Choose a VCS: Use Git, GitHub, GitLab, Bitbucket, etc.
- Create a Repository: Create a new repository or use an existing one.
- Commit Code: Ensure your code is committed to the repository.

Step 3. Install Required Plugins in Jenkins

- Git Plugin: For connecting Jenkins with your Git repository.
- Pipeline Plugin: For defining Jenkins pipelines.

- Other Plugins: Depending on your project requirements (e.g., Maven, Docker, NodeJS).

Step 4. Create a New Jenkins Pipeline Job

- New Item: From the Jenkins dashboard, click on ?New Item?.
- Pipeline: Select ?Pipeline? and give your job a name.
- OK: Click ?OK? to create the job.

Step 5. Configure the Pipeline Job

- Pipeline Script from SCM: Under the ?Pipeline? section, select ?Pipeline script from SCM?.
- SCM: Select ?Git? and provide the repository URL.
- Branch Specifier: Specify which branch to build (e.g., */main).

Step 6. Define the Pipeline Script

- Jenkinsfile: Create a Jenkinsfile in your repository root. This file will define the stages of your CI/CD pipeline.

7. Trigger Builds Automatically

- Webhooks: Set up webhooks in your VCS to trigger Jenkins builds automatically on commits.
- Poll SCM: Alternatively, you can configure Jenkins to poll the SCM periodically.

8. Monitor and Manage Builds

- Build Dashboard: Monitor build status from the Jenkins dashboard.
- Logs: Check build logs for details on failures and successes.
- Notifications: Configure email or chat notifications for build results.

9. Secure Your Pipeline

- Access Control: Set up proper access controls in Jenkins to ensure only authorized users can modify the pipeline.
- Secrets Management: Use Jenkins credentials plugin to manage sensitive information securely.

10. Optimize and Scale

- Parallel Builds: Configure Jenkins to run builds in parallel to speed up the process.
- Distributed Builds: Set up Jenkins agents on multiple nodes to distribute the load.
- Pipeline as Code: Keep your pipeline configuration in the repository (Jenkinsfile) to version control your CI/CD process.

Pipeline Orchestration

Pipeline orchestration in system design involves managing and automating the various stages of a CI/CD pipeline to ensure smooth and efficient delivery of software. Orchestration ensures that each stage of the pipeline, from code commit to production deployment, is executed in the correct order and according to predefined rules. Here's an in-depth look at pipeline orchestration in system design:

Key Components of Pipeline Orchestration

- Pipeline Definition: Declarative Pipelines: Use a domain-specific language (DSL) to define the stages and steps of the pipeline in code. This makes the pipeline easy to version control and modify. Scripted Pipelines: Use general-purpose scripting languages to define complex workflows and custom logic.
- Declarative Pipelines: Use a domain-specific language (DSL) to define the stages and steps of the pipeline in code. This makes the pipeline easy to version control and modify.
- Scripted Pipelines: Use general-purpose scripting languages to define complex workflows and custom logic.
- Stage Management: Sequential Stages: Define stages that execute one after another, ensuring a linear flow from development to deployment. Parallel Stages: Execute multiple stages simultaneously to optimize the use of resources and reduce overall pipeline execution time.
- Sequential Stages: Define stages that execute one after another, ensuring a linear flow from development to deployment.
- Parallel Stages: Execute multiple stages simultaneously to optimize the use of resources and

reduce overall pipeline execution time.

- Conditionals and Branching:Conditional Execution:Execute stages or steps based on specific conditions, such as the branch name, environment variables, or the results of previous steps.Branching:Define different workflows for different branches (e.g., feature branches, main branch) to manage different stages of development and release.
- Conditional Execution:Execute stages or steps based on specific conditions, such as the branch name, environment variables, or the results of previous steps.
- Branching:Define different workflows for different branches (e.g., feature branches, main branch) to manage different stages of development and release.
- Pipeline Triggers:Event-Based Triggers:Start the pipeline based on events such as code commits, pull requests, or manual triggers.Scheduled Triggers:Execute pipelines at scheduled intervals to perform tasks like nightly builds or periodic testing.
- Event-Based Triggers:Start the pipeline based on events such as code commits, pull requests, or manual triggers.
- Scheduled Triggers:Execute pipelines at scheduled intervals to perform tasks like nightly builds or periodic testing.
- Artifact Management:Build Artifacts:Manage the outputs of the build process (e.g., compiled binaries, Docker images) and ensure they are passed correctly between stages.Artifact Storage:Use artifact repositories (e.g., JFrog Artifactory, Nexus) to store and manage build artifacts.
- Build Artifacts:Manage the outputs of the build process (e.g., compiled binaries, Docker images) and ensure they are passed correctly between stages.
- Artifact Storage:Use artifact repositories (e.g., JFrog Artifactory, Nexus) to store and manage build artifacts.
- Environment Management:Environment Variables:Define and manage environment variables that are required for different stages of the pipeline.Infrastructure as Code (IaC):Use tools like Terraform or Ansible to provision and manage infrastructure as part of the pipeline.
- Environment Variables:Define and manage environment variables that are required for different

stages of the pipeline.

- Infrastructure as Code (IaC): Use tools like Terraform or Ansible to provision and manage infrastructure as part of the pipeline.

Key Components of Pipeline Orchestration

- Declarative Pipelines: Use a domain-specific language (DSL) to define the stages and steps of the pipeline in code. This makes the pipeline easy to version control and modify.
- Scripted Pipelines: Use general-purpose scripting languages to define complex workflows and custom logic.

Key Components of Pipeline Orchestration

- Sequential Stages: Define stages that execute one after another, ensuring a linear flow from development to deployment.
- Parallel Stages: Execute multiple stages simultaneously to optimize the use of resources and reduce overall pipeline execution time.

Key Components of Pipeline Orchestration

- Conditional Execution: Execute stages or steps based on specific conditions, such as the branch name, environment variables, or the results of previous steps.
- Branching: Define different workflows for different branches (e.g., feature branches, main branch) to manage different stages of development and release.

Key Components of Pipeline Orchestration

- Event-Based Triggers: Start the pipeline based on events such as code commits, pull requests, or manual triggers.
- Scheduled Triggers: Execute pipelines at scheduled intervals to perform tasks like nightly builds or periodic testing.

Key Components of Pipeline Orchestration

- **Build Artifacts:**Manage the outputs of the build process (e.g., compiled binaries, Docker images) and ensure they are passed correctly between stages.
- **Artifact Storage:**Use artifact repositories (e.g., JFrog Artifactory, Nexus) to store and manage build artifacts.

Key Components of Pipeline Orchestration

- **Environment Variables:**Define and manage environment variables that are required for different stages of the pipeline.
- **Infrastructure as Code (IaC):**Use tools like Terraform or Ansible to provision and manage infrastructure as part of the pipeline.

Security in CI/CD Pipelines

Ensuring security in CI/CD pipelines is critical to protect the software development lifecycle from vulnerabilities and threats. Implementing security best practices at each stage of the CI/CD pipeline helps safeguard code, build artifacts, and deployment environments. Here are key considerations and best practices for securing CI/CD pipelines:

1. Code Security

- **Code Reviews:**Enforce peer reviews for all code changes. Tools like GitHub and GitLab provide features for mandatory code reviews before merging.
- **Static Code Analysis:**Use tools like SonarQube, Checkmarx, or Snyk to analyze code for vulnerabilities and coding standards during the build stage.
- **Secrets Management:**Store sensitive information such as API keys, passwords, and tokens in secure vaults like HashiCorp Vault, AWS Secrets Manager, or Azure Key Vault. Avoid hardcoding secrets in the codebase.

2. Build Security

- **Build Isolation:**Run builds in isolated environments (e.g., containers or virtual machines) to prevent cross-contamination and unauthorized access.

- **Dependency Management:**Use tools like OWASP Dependency-Check, Snyk, or WhiteSource to scan dependencies for known vulnerabilities. Regularly update dependencies to their latest secure versions.
- **Signed Artifacts:**Sign build artifacts cryptographically to ensure their integrity and authenticity. Verify signatures before deploying artifacts to production.

3. Environment Security

- **Infrastructure as Code (IaC):**Use IaC tools like Terraform, Ansible, or CloudFormation to define and manage infrastructure securely. Store IaC configurations in version-controlled repositories.
- **Environment Segmentation:**Isolate environments (development, staging, production) to minimize the risk of unauthorized access and data leakage. Use network segmentation and firewall rules to control access between environments.
- **Access Controls:**Implement role-based access control (RBAC) to restrict access to CI/CD tools and environments. Use principles of least privilege to grant minimum necessary permissions to users and service accounts.

4. Pipeline Security

- **Secure CI/CD Tools:**Ensure the CI/CD tool (e.g., Jenkins, GitLab CI/CD, CircleCI) is up to date with security patches. Regularly audit configurations and permissions.
- **Pipeline Hardening:**Enforce strict access controls on pipeline configurations and scripts. Use encrypted communication channels (e.g., HTTPS, SSH) for data transmission.
- **Third-Party Integrations:**Evaluate and limit the use of third-party plugins and integrations. Ensure they come from trusted sources and are regularly updated.

5. Runtime Security

- **Container Security:**Use container security tools like Aqua, Twistlock (Palo Alto Prisma), or Clair to scan container images for vulnerabilities. Follow best practices for container security, such as running containers with non-root users and minimizing the attack surface.

- Continuous Monitoring: Implement monitoring and logging for CI/CD pipeline activities. Use tools like ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, or Prometheus to monitor pipeline logs and detect anomalies.
- Incident Response: Develop and maintain an incident response plan for CI/CD pipeline security breaches. Regularly conduct drills and update the plan based on lessons learned.

6. Compliance and Governance

- Policy Enforcement: Use policy-as-code tools like Open Policy Agent (OPA) or Sentinel to enforce security and compliance policies across the CI/CD pipeline.
- Audit Trails: Maintain detailed audit logs of pipeline activities, including code commits, build processes, and deployment actions. Ensure logs are tamper-evident and stored securely.
- Compliance Checks: Integrate compliance checks into the CI/CD pipeline to ensure adherence to industry standards and regulations (e.g., GDPR, HIPAA, PCI-DSS).

Scaling CI/CD Pipelines

Scaling CI/CD pipelines is essential for supporting larger teams, handling increased load, and ensuring high availability. Here are key considerations and best practices for scaling CI/CD pipelines effectively:

Scaling CI/CD Pipelines

- Infrastructure Scalability: Horizontal Scaling: Add more CI/CD servers or agents to handle additional load. Tools like Jenkins, GitLab CI, and CircleCI support adding multiple agents to distribute the workload. Containerization: Use containers to run CI/CD jobs in isolated, reproducible environments. Tools like Docker and Kubernetes can help manage and scale containerized workloads. Cloud Services: Leverage cloud-based CI/CD services (e.g., AWS CodePipeline, Azure DevOps, GitHub Actions) to automatically scale infrastructure based on demand.
- Horizontal Scaling: Add more CI/CD servers or agents to handle additional load. Tools like Jenkins, GitLab CI, and CircleCI support adding multiple agents to distribute the workload.

- Containerization: Use containers to run CI/CD jobs in isolated, reproducible environments. Tools like Docker and Kubernetes can help manage and scale containerized workloads.
- Cloud Services: Leverage cloud-based CI/CD services (e.g., AWS CodePipeline, Azure DevOps, GitHub Actions) to automatically scale infrastructure based on demand.
- Parallel Execution:
 - Parallel Jobs: Configure pipelines to run multiple jobs in parallel. This can significantly reduce the time required to complete the pipeline, especially for build and test stages.
 - Matrix Builds: Use matrix builds to run the same set of tests or builds across different environments or configurations (e.g., multiple OS versions, different programming language versions).
- Parallel Jobs: Configure pipelines to run multiple jobs in parallel. This can significantly reduce the time required to complete the pipeline, especially for build and test stages.
- Matrix Builds: Use matrix builds to run the same set of tests or builds across different environments or configurations (e.g., multiple OS versions, different programming language versions).
- Pipeline Optimization:
 - Caching: Implement caching mechanisms to reuse dependencies and artifacts between builds. This reduces the time spent on downloading and installing dependencies.
 - Incremental Builds: Configure the pipeline to only build and test changed components rather than the entire project. This can be achieved using tools like Bazel or Gradle's incremental build feature.
 - Pipeline as Code: Define CI/CD pipelines as code to version control and easily replicate pipeline configurations across different projects and environments.
- Caching: Implement caching mechanisms to reuse dependencies and artifacts between builds. This reduces the time spent on downloading and installing dependencies.
- Incremental Builds: Configure the pipeline to only build and test changed components rather than the entire project. This can be achieved using tools like Bazel or Gradle's incremental build feature.
- Pipeline as Code: Define CI/CD pipelines as code to version control and easily replicate pipeline configurations across different projects and environments.
- Load Balancing:
 - Load Balancers: Use load balancers to distribute incoming CI/CD job requests evenly across multiple servers or agents. This ensures no single server is overwhelmed.
 - Distributed

Builds:Use distributed build systems that can split large builds into smaller tasks and run them across multiple nodes.

- **Load Balancers:**Use load balancers to distribute incoming CI/CD job requests evenly across multiple servers or agents. This ensures no single server is overwhelmed.

- **Distributed Builds:**Use distributed build systems that can split large builds into smaller tasks and run them across multiple nodes.

- **Monitoring and Logging:****Centralized Logging:**Implement centralized logging solutions (e.g., ELK Stack, Splunk) to aggregate logs from all CI/CD components. This helps in monitoring pipeline health and troubleshooting issues.**Performance Monitoring:**Use monitoring tools (e.g., Prometheus, Grafana) to track the performance of CI/CD pipelines. Monitor metrics like job duration, resource utilization, and failure rates.

- **Centralized Logging:**Implement centralized logging solutions (e.g., ELK Stack, Splunk) to aggregate logs from all CI/CD components. This helps in monitoring pipeline health and troubleshooting issues.

- **Performance Monitoring:**Use monitoring tools (e.g., Prometheus, Grafana) to track the performance of CI/CD pipelines. Monitor metrics like job duration, resource utilization, and failure rates.

- **High Availabilityand Fault Tolerance:****Redundancy:**Set up redundant CI/CD servers or agents to ensure high availability. Use techniques like active-active or active-passive configurations.**Automated Failover:**Implement automated failover mechanisms to switch to backup servers in case of failures. Tools like Kubernetes can help manage failover for containerized workloads.

- **Redundancy:**Set up redundant CI/CD servers or agents to ensure high availability. Use techniques like active-active or active-passive configurations.

- **Automated Failover:**Implement automated failover mechanisms to switch to backup servers in case of failures. Tools like Kubernetes can help manage failover for containerized workloads.

Scaling CI/CD Pipelines

- Horizontal Scaling: Add more CI/CD servers or agents to handle additional load. Tools like Jenkins, GitLab CI, and CircleCI support adding multiple agents to distribute the workload.
- Containerization: Use containers to run CI/CD jobs in isolated, reproducible environments. Tools like Docker and Kubernetes can help manage and scale containerized workloads.
- Cloud Services: Leverage cloud-based CI/CD services (e.g., AWS CodePipeline, Azure DevOps, GitHub Actions) to automatically scale infrastructure based on demand.

Scaling CI/CD Pipelines

- Parallel Jobs: Configure pipelines to run multiple jobs in parallel. This can significantly reduce the time required to complete the pipeline, especially for build and test stages.
- Matrix Builds: Use matrix builds to run the same set of tests or builds across different environments or configurations (e.g., multiple OS versions, different programming language versions).

Scaling CI/CD Pipelines

- Caching: Implement caching mechanisms to reuse dependencies and artifacts between builds. This reduces the time spent on downloading and installing dependencies.
- Incremental Builds: Configure the pipeline to only build and test changed components rather than the entire project. This can be achieved using tools like Bazel or Gradle's incremental build feature.
- Pipeline as Code: Define CI/CD pipelines as code to version control and easily replicate pipeline configurations across different projects and environments.

Scaling CI/CD Pipelines

- Load Balancers: Use load balancers to distribute incoming CI/CD job requests evenly across multiple servers or agents. This ensures no single server is overwhelmed.
- Distributed Builds: Use distributed build systems that can split large builds into smaller tasks and run them across multiple nodes.

Scaling CI/CD Pipelines

- **Centralized Logging:**Implement centralized logging solutions (e.g., ELK Stack, Splunk) to aggregate logs from all CI/CD components. This helps in monitoring pipeline health and troubleshooting issues.
- **Performance Monitoring:**Use monitoring tools (e.g., Prometheus, Grafana) to track the performance of CI/CD pipelines. Monitor metrics like job duration, resource utilization, and failure rates.

Scaling CI/CD Pipelines

- **Redundancy:**Set up redundant CI/CD servers or agents to ensure high availability. Use techniques like active-active or active-passive configurations.
- **Automated Failover:**Implement automated failover mechanisms to switch to backup servers in case of failures. Tools like Kubernetes can help manage failover for containerized workloads.

Common Challenges with CI/CD Pipeline and How to Avoid Them

Continuous Integration/Continuous Deployment (CI/CD) pipelines are powerful tools for automating software delivery processes, but they come with their own set of challenges. Here are some common challenges with CI/CD pipelines and how to avoid or mitigate them:

Common Challenges with CI/CD Pipeline and How to Avoid Them

- **Complexity of Pipeline Configuration:****Challenge:**CI/CD pipelines can become complex, especially in large projects with multiple stages and environments.**Solution:**Use Infrastructure as Code (IaC) tools like Terraform or CloudFormation to manage pipeline configurations. Break down pipelines into smaller, reusable components. Use version control for pipeline configurations to track changes and facilitate collaboration.
- **Challenge:**CI/CD pipelines can become complex, especially in large projects with multiple stages and environments.
- **Solution:**Use Infrastructure as Code (IaC) tools like Terraform or CloudFormation to manage pipeline configurations. Break down pipelines into smaller, reusable components. Use version

control for pipeline configurations to track changes and facilitate collaboration.

- Integration Issues:Challenge:Ensuring that various tools and systems integrate seamlessly within the pipeline (e.g., version control, testing frameworks, deployment targets).Solution:Regularly update and test integrations between tools. Use standardized APIs and plugins provided by CI/CD platforms. Automate integration testing as part of your pipeline.

- Challenge:Ensuring that various tools and systems integrate seamlessly within the pipeline (e.g., version control, testing frameworks, deployment targets).

- Solution:Regularly update and test integrations between tools. Use standardized APIs and plugins provided by CI/CD platforms. Automate integration testing as part of your pipeline.

- Slow Build/Test Execution:Challenge:Long build and test times can delay feedback and deployment.Solution:Optimize build processes by parallelizing tasks and leveraging caching mechanisms. Use containerization (e.g., Docker) to create consistent environments for builds and tests. Implement incremental builds to only build/test what has changed.

- Challenge:Long build and test times can delay feedback and deployment.

- Solution:Optimize build processes by parallelizing tasks and leveraging caching mechanisms. Use containerization (e.g., Docker) to create consistent environments for builds and tests. Implement incremental builds to only build/test what has changed.

- Maintaining Pipeline Reliability:Challenge:Ensuring that the CI/CD pipeline is reliable and consistent in delivering builds.Solution:Monitor pipeline performance and reliability metrics. Implement automated retries and notifications for failed builds. Regularly review and update pipeline configurations and dependencies.

- Challenge:Ensuring that the CI/CD pipeline is reliable and consistent in delivering builds.

- Solution:Monitor pipeline performance and reliability metrics. Implement automated retries and notifications for failed builds. Regularly review and update pipeline configurations and dependencies.

- Security Concerns:Challenge:Vulnerabilities in pipeline components or improper handling of credentials can lead to security breaches.Solution:Use secrets management tools (e.g., HashiCorp

Vault, AWS Secrets Manager) to securely store and access credentials. Implement least privilege access controls for pipeline components. Conduct regular security audits and vulnerability assessments.

- Challenge: Vulnerabilities in pipeline components or improper handling of credentials can lead to security breaches.

- Solution: Use secrets management tools (e.g., HashiCorp Vault, AWS Secrets Manager) to securely store and access credentials. Implement least privilege access controls for pipeline components. Conduct regular security audits and vulnerability assessments.

- Scaling Challenges: Challenge: Scaling CI/CD pipelines to handle increased workload and larger teams. Solution: Use scalable CI/CD platforms that support distributed builds and parallel execution. Monitor resource utilization and adjust pipeline configurations accordingly. Implement pipeline as code practices to easily replicate and scale pipelines.

- Challenge: Scaling CI/CD pipelines to handle increased workload and larger teams.

- Solution: Use scalable CI/CD platforms that support distributed builds and parallel execution. Monitor resource utilization and adjust pipeline configurations accordingly. Implement pipeline as code practices to easily replicate and scale pipelines.

- Lack of Testing Coverage: Challenge: Inadequate testing can result in bugs reaching production. Solution: Implement a comprehensive testing strategy (unit, integration, regression, performance). Automate testing at every stage of the pipeline. Integrate testing frameworks with CI/CD tools for seamless execution and reporting.

- Challenge: Inadequate testing can result in bugs reaching production.

- Solution: Implement a comprehensive testing strategy (unit, integration, regression, performance). Automate testing at every stage of the pipeline. Integrate testing frameworks with CI/CD tools for seamless execution and reporting.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge: CI/CD pipelines can become complex, especially in large projects with multiple stages

and environments.

- Solution: Use Infrastructure as Code (IaC) tools like Terraform or CloudFormation to manage pipeline configurations. Break down pipelines into smaller, reusable components. Use version control for pipeline configurations to track changes and facilitate collaboration.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge: Ensuring that various tools and systems integrate seamlessly within the pipeline (e.g., version control, testing frameworks, deployment targets).
- Solution: Regularly update and test integrations between tools. Use standardized APIs and plugins provided by CI/CD platforms. Automate integration testing as part of your pipeline.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge: Long build and test times can delay feedback and deployment.
- Solution: Optimize build processes by parallelizing tasks and leveraging caching mechanisms. Use containerization (e.g., Docker) to create consistent environments for builds and tests. Implement incremental builds to only build/test what has changed.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge: Ensuring that the CI/CD pipeline is reliable and consistent in delivering builds.
- Solution: Monitor pipeline performance and reliability metrics. Implement automated retries and notifications for failed builds. Regularly review and update pipeline configurations and dependencies.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge: Vulnerabilities in pipeline components or improper handling of credentials can lead to security breaches.
- Solution: Use secrets management tools (e.g., HashiCorp Vault, AWS Secrets Manager) to securely store and access credentials. Implement least privilege access controls for pipeline components. Conduct regular security audits and vulnerability assessments.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge:Scaling CI/CD pipelines to handle increased workload and larger teams.
- Solution:Use scalable CI/CD platforms that support distributed builds and parallel execution. Monitor resource utilization and adjust pipeline configurations accordingly. Implement pipeline as code practices to easily replicate and scale pipelines.

Common Challenges with CI/CD Pipeline and How to Avoid Them

- Challenge:Inadequate testing can result in bugs reaching production.
- Solution:Implement a comprehensive testing strategy (unit, integration, regression, performance). Automate testing at every stage of the pipeline. Integrate testing frameworks with CI/CD tools for seamless execution and reporting.

Conclusion

In conclusion, designing a robust CI/CD pipeline is crucial for efficient software delivery. By overcoming challenges like complexity in configuration, integration issues, and ensuring fast and reliable builds, teams can streamline the development process. Security measures and scalable practices also play key roles in maintaining pipeline integrity. Emphasizing comprehensive testing and continuous optimization ensures high-quality software reaches production faster. Ultimately, a well-designed CI/CD pipeline enhances collaboration, reduces errors, and supports agile development, enabling teams to deliver value to users consistently and reliably.

Conclusion

Conclusion

Become a System Design expert and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Conclusion

Step into the Three 90 Challenge ? 90 days to push limits, break barriers, and become the best

version of yourself! Don't miss your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Conclusion

- System Design

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos

- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- [DSA](#)
- [Data Structures](#)
- [Algorithms](#)
- [DSA for Beginners](#)
- [Basic DSA Problems](#)
- [DSA Roadmap](#)
- [DSA Interview Questions](#)
- [Competitive Programming](#)

Similar Reads

- [Data Science & ML](#)
- [Data Science With Python](#)
- [Data Science For Beginner](#)
- [Machine Learning](#)

- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD

- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL

- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management

- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course

- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT

- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Basics of Computer Network

- Basics of Computer Networking
- Introduction to basic Networking Terminology
- Goals of Networks
- Basic Characteristics of Computer Networks
- Challenges of Computer Network
- Physical Components of Computer Network

Network Hardware and Software

- Types of Computer Networks
- LAN Full Form - Local area network
- How to Set Up a LAN Network?
- MAN Full Form in Computer Networking
- MAN Full Form
- WAN Full Form
- Introduction of Internetworking
- Difference between Internet, Intranet and Extranet
- Protocol Hierarchies in Computer Network
- Network Devices (Hub, Repeater, Bridge, Switch, Router, Gateways and Brouter)
- Introduction of a Router
- Introduction of Gateways
- What is a Network Switch and How Does it Work?

Network Topology

- Types of Network Topology
- Difference between Physical and Logical Topology

OSI Model

- What is OSI Model? - Layers of OSI Model
- Physical Layer in OSI Model
- Data Link Layer in OSI Model
- Session Layer in OSI model
- Presentation Layer in OSI model
- Application Layer in OSI Model

Protocols

- Protocol and Standard in Computer Networks
- Examples of Data Link Layer Protocols

TCP/IP Model

- TCP/IP Model
- TCP/IP Ports and Its Applications
- What is TCP (Transmission Control Protocol)?
- TCP 3-Way Handshake Process
- Services and Segment structure in TCP
- TCP Connection Establishment
- TCP Connection Termination
- TCP Timers
- Fast Recovery Technique For Loss Recovery in TCP
- Difference Between OSI Model and TCP/IP Model

Medium Access Control

- MAC Full Form - Media Access Control
- Channel Allocation Problem in Computer Network
- Multiple Access Protocols in Computer Network
- Carrier Sense Multiple Access (CSMA)

- Collision Detection in CSMA/CD
- Controlled Access Protocols in Computer Network

SLIDING WINDOW PROTOCOLS

- Stop and Wait ARQ
- Sliding Window Protocol | Set 3 (Selective Repeat)
- Piggybacking in Computer Networks

IP Addressing

- What is IPv4?
- What is IPv6?
- Introduction of Classful IP Addressing
- Classless Addressing in IP Addressing
- Classful vs Classless Addressing
- Classless Inter Domain Routing (CIDR)
- Supernetting in Network Layer
- Introduction To Subnetting
- Difference between Subnetting and Supernetting

Routing

- Types of Routing
- Difference between Static and Dynamic Routing
- Unicast Routing - Link State Routing
- Distance Vector Routing (DVR) Protocol
- Fixed and Flooding Routing algorithms
- Introduction of Firewall in Computer Network

Congestion Control Algorithms

- Congestion Control in Computer Networks

- Congestion Control techniques in Computer Networks
- Computer Network | Leaky bucket algorithm
- TCP Congestion Control

Network Switching

- Circuit Switching in Computer Network
- Message switching techniques
- Packet Switching and Delays in Computer Network
- Differences Between Virtual Circuits and Datagram Networks

Application Layer:DNS

- Domain Name System (DNS)
- Details on DNS
- Introduction to Electronic Mail
- E-Mail Format
- World Wide Web (WWW)
- HTTP Full Form - Hypertext Transfer Protocol
- Streaming Stored Video
- What is a Content Distribution Network and how does it work?

CN Interview Questions

- Top 50 Plus Networking Interview Questions and Answers for 2024
- Top 50 TCP/IP Interview Questions and Answers 2025
- Top 50 IP Addressing Interview Questions and Answers
- Last Minute Notes for Computer Networks
- Computer Network - Cheat Sheet

CN MCQ

- Network Layer

- Transport Layer
- Application Layer

Computer Network Tutorial

-
-
-

Computer Network Tutorial

A Computer Network is a system where two or more devices are linked together to share data, resources, and information. These networks can range from simple setups, like connecting two devices in your home, to massive global systems, like the Internet.

Computer Network Tutorial

Computer networks are an important part of our daily lives. They allow devices like computers, smartphones, and tablets to connect and communicate with each other. Whether you're browsing the internet, sending an email, or streaming a video, it all happens because of computer networks.

Computer Network Tutorial

This complete Computer Networks tutorial will guide you through the basics of computer networks, explaining how they work, the different types of networks, and their importance in today's world. Also, you'll get to know other fundamental and advanced Computer Networking concepts like OSI Model, IP Addressing, Subnetting, Network Security, etc.

Computer Network Tutorial

So, let's get started.

Introduction

- Basics of Computer Networks
- Characteristics of Computer Network

- Types of Network

Network Architecture

- Layered Architecture
- Peer-to-Peer Network
- Client Server Network

Network Topologies

- Mesh Topology
- Bus Topology
- Ring Topology
- Star Topology
- Tree Topology
- Hybrid Topology

Networking Devices

- Routers
- Switches
- Hubs
- Access Points
- Modems

Network Models

- OSI Model
- TCP/IP Model

Physical Layer

- Physical Layer
- Transmission Media

- Switching
- Types of Switching Techniques (Circuit,Message,packet)
- VLANs (Virtual Local Area Networks)
- STP (Spanning Tree Protocol)
- Link Aggregation

Data Link Layer

- Data Link Layer
- Framing in Data Link Layer
- Error Detection Techniques
- Error Correction Techniques
- Error Control in Data Link Layer
- Flow Control in Data Link Layer
- Stop and wait ARQ
- Sliding Window Protocol- Go Back N
- Sliding Window Protocol- Selective Repeat

Network Layer

- Network Layer
- Classful Network Addressing
- Classless Network Addressing
- What is an IP address?
- IPv4 Header Format
- IPv4 vs IPv6
- Private vs Public IP addresses

Subnetting

- Subnetting basics

- Subnet masks
- Calculate network, broadcast, and host addresses
- Variable Length Subnet Masking (VLSM)
- Advanced VLSM
- Supernetting
- Subnetting large networks

Routing

- What is Routing?
- Static vs Dynamic Routing
- Routing Protocols: RIP, OSPF, EIGRP, BGP, MPLS
- Network Address Translation(NAT)
- Default Gateway

Network Layer Protocols

- ARP
- RARP
- DHCP

Transport Layer

- Transport Layer
- Transport Layer Protocol-TCP
- TCP 3-Way Handshake Process
- Congestion Control in TCP
- Transport Layer Protocol-UPD

Session and Presentation Layer

- Session Layer
- Presentation Layer

Application Layer

- Application Layer
- Client-Server Model
- DNS
- FTP
- SMTP
- SNMP
- HTTP

QoS (Quality of Service)

- What is QoS and Multimedia?
- Techniques for achieving QoS
- Token Bucket
- Leaky Bucket

QoS (Quality of Service)

Quickly review our Last Minute Notes on Computer Networks and test your knowledge with our practice quiz.

QoS (Quality of Service)

Dreaming of M.Tech in IIT? Get AIR under 100 with our GATE 2026 CSE & DA courses! Get flexible weekday/weekend options, live mentorship, and mock tests. Access exclusive features like All India Mock Tests, and Doubt Solving? your GATE success starts now!

QoS (Quality of Service)

- Computer Networks
- GATE CS

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++

- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System

- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics

- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates

- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD

- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course

- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

C++ Overview

- Introduction to C++ Programming Language
- Features of C++
- History of C++
- Interesting Facts about C++
- Setting up C++ Development Environment
- Difference between C and C++

C++ Basics

- Writing First C++ Program - Hello World Example
- C++ Basic Syntax
- C++ Comments
- Tokens in C
- C++ Keywords
- Difference between Keyword and Identifier in C

C++ Variables and Constants

- C++ Variables
- Constants in C
- Scope of Variables in C++
- Storage Classes in C++ with Examples
- Static Keyword in C++

C++ Data Types and Literals

- C++ Data Types
- Literals in C
- Derived Data Types in C++
- User Defined Data Types in C++

- Data Type Ranges and Their Macros in C++
- C++ Type Modifiers
- Type Conversion in C++
- Casting Operators in C++

C++ Operators

- Operators in C++
- C++ Arithmetic Operators
- Unary operators in C
- Bitwise Operators in C
- Assignment Operators in C
- C++ sizeof Operator
- Scope Resolution Operator in C++

C++ Input/Output

- Basic Input / Output in C++
- cin in C++
- cout in C++
- Standard Error Stream Object - cerr in C++
- Manipulators in C++

C++ Control Statements

- Decision Making in C (if , if..else, Nested if, if-else-if)
- C++ if Statement
- C++ if else Statement
- C++ if else if Ladder
- Switch Statement in C++
- Jump statements in C++

- C++ Loops
- for Loop in C++
- Range-Based for Loop in C++
- C++ While Loop
- C++ do while Loop

C++ Functions

- Functions in C++
- return Statement in C++
- Parameter Passing Techniques in C
- Difference Between Call by Value and Call by Reference in C
- Default Arguments in C++
- Inline Functions in C++
- Lambda expression in C++

C++ Pointers and References

- Pointers and References in C++
- C++ Pointers
- Dangling, Void , Null and Wild Pointers in C
- Applications of Pointers in C
- Understanding nullptr in C++
- References in C++
- Can References Refer to Invalid Location in C++?
- Pointers vs References in C++
- Passing By Pointer vs Passing By Reference in C++
- When do we pass arguments by pointer?

C++ Arrays

- Variable Length Arrays (VLAs) in C
- Pointer to an Array | Array Pointer
- How to print size of array parameter in C++?
- Pass Array to Functions in C
- What is Array Decay in C++? How can it be prevented?

C++ Strings

- Strings in C++
- std::string class in C++
- Array of Strings in C++ - 5 Different Ways to Create
- String Concatenation in C++
- Tokenizing a string in C++
- Substring in C++

C++ Structures and Unions

- Structures, Unions and Enumerations in C++
- Structures in C++
- C++ - Pointer to Structure
- Self Referential Structures
- Difference Between C Structures and C++ Structures
- Enumeration in C++
- typedef in C++
- Array of Structures vs Array within a Structure in C

C++ Dynamic Memory Management

- Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()
- new and delete Operators in C++ For Dynamic Memory
- new vs malloc() and free() vs delete in C++

- What is Memory Leak? How can we avoid?
- Difference between Static and Dynamic Memory Allocation in C

C++ Object-Oriented Programming

- Object Oriented Programming in C++
- C++ Classes and Objects
- Access Modifiers in C++
- Friend Class and Function in C++
- Constructors in C++
- Default Constructors in C++
- Copy Constructor in C++
- Destructors in C++
- Private Destructor in C++
- When is a Copy Constructor Called in C++?
- Shallow Copy and Deep Copy in C++
- When Should We Write Our Own Copy Constructor in C++?
- Does C++ compiler create default constructor when we write our own?
- C++ Static Data Members
- Static Member Function in C++
- 'this' pointer in C++
- Scope Resolution Operator vs this pointer in C++
- Local Classes in C++
- Nested Classes in C++
- Enum Classes in C++ and Their Advantage over Enum DataType
- Difference Between Structure and Class in C++
- Why C++ is partially Object Oriented Language?

C++ Encapsulation and Abstraction

- Encapsulation in C++
- Abstraction in C++
- Difference between Abstraction and Encapsulation in C++

C++ Polymorphism

- C++ Polymorphism
- Function Overriding in C++
- Virtual Functions and Runtime Polymorphism in C++
- Difference between Inheritance and Polymorphism

C++ Function Overloading

- Function Overloading in C++
- Constructor Overloading in C++
- Functions that cannot be overloaded in C++
- Function overloading and const keyword
- Function Overloading and Return Type in C++
- Function Overloading and float in C++
- Can main() be overloaded in C++?
- Function Overloading vs Function Overriding in C++
- Advantages and Disadvantages of Function Overloading in C++

C++ Operator Overloading

- Operator Overloading in C++
- Types of Operator Overloading in C++
- Functors in C++
- What are the Operators that Can be and Cannot be Overloaded in C++?

C++ Inheritance

- Inheritance in C++

- C++ Inheritance Access
- Multiple Inheritance in C++
- C++ Hierarchical Inheritance
- C++ Multilevel Inheritance
- Constructor in Multiple Inheritance in C++
- Inheritance and Friendship in C++
- Does overloading work with Inheritance?
- Difference between Inheritance and Polymorphism

C++ Virtual Functions

- Virtual Function in C++
- Virtual Functions in Derived Classes in C++
- Default Arguments and Virtual Function in C++
- Can Virtual Functions be Inlined in C++?
- Virtual Destructor
- Advanced C++ | Virtual Constructor
- Advanced C++ | Virtual Copy Constructor
- Pure Virtual Functions and Abstract Classes in C++
- Pure Virtual Destructor in C++
- Can Static Functions Be Virtual in C++?
- RTTI (Run-Time Type Information) in C++
- Can Virtual Functions be Private in C++?

C++ Exception Handling

- Exception Handling in C++
- Exception Handling using classes in C++
- Stack Unwinding in C++
- User-defined Custom Exception with class in C++

C++ Files and Streams

- File Handling through C++ Classes
- I/O Redirection in C++

C++ Templates

- Templates in C++ with Examples
- Using Keyword in C++ STL

C++ Standard Template Library (STL)

- C++ Standard Template Library (STL)
- Containers in C++ STL (Standard Template Library)
- Introduction to Iterators in C++
- Algorithm Library | C++ Magicians STL Algorithm
- Functors in C++

C++ Preprocessors

- C Preprocessors
- C Preprocessor Directives
- #include in C
- C Preprocessors
- C Preprocessor Directives
- Difference between Preprocessor Directives and Function Templates in C++

C++ Namespace

- Namespace in C++ | Set 1 (Introduction)
- namespace in C++ | Set 2 (Extending namespace and Unnamed namespace)
- Namespace in C++ | Set 3 (Accessing, creating header, nesting and aliasing)
- C++ Inline Namespaces and Usage of the "using" Directive Inside Namespaces

Advanced C++

- Multithreading in C++
- Smart Pointers in C++
- auto_ptr vs unique_ptr vs shared_ptr vs weak_ptr in C++
- Type of 'this' Pointer in C++
- "delete this" in C++
- Passing a Function as a Parameter in C++
- Signal Handling in C++
- Generics in C++

C vs C++

- Difference between C and C++
- Difference between C++ and Objective C
- Write a C program that won't compile in C++
- Write a program that produces different results in C and C++
- How does 'void*' differ in C and C++?
- Type Difference of Character Literals in C and C++
- Difference Between C Structures and C++ Structures
- Cin-Cout vs Scanf-Printf

C++ vs Java

- Similarities and Difference between Java and C++
- Comparison of Inheritance in C++ and Java
- Static Keyword in C++
- How Does Default Virtual Behavior Differ in C++ and Java?
- Comparison of Exception Handling in C++ and Java
- Foreach in C++ and Java
- Templates in C++ vs Generics in Java

- Floating Point Operations & Associativity in C, C++ and Java

Competitive Programming in C++

- Competitive Programming - A Complete Guide
- C++ tricks for competitive programming (for C++ 11)
- Writing C/C++ code efficiently in Competitive programming
- Why C++ is best for Competitive Programming?
- Test Case Generation | Set 1 (Random Numbers, Arrays and Matrices)
- Fast I/O for Competitive Programming
- Setting up Sublime Text for C++ Competitive Programming Environment
- How to setup Competitive Programming in Visual Studio Code for C++
- Which C++ libraries are useful for competitive programming?
- Common mistakes to be avoided in Competitive Programming in C++ | Beginners

C++ Interview Questions

- C++ Interview Questions and Answers (2025)
- Top C++ STL Interview Questions and Answers
- 30 OOPs Interview Questions and Answers [2025 Updated]
- Top C++ Exception Handling Interview Questions and Answers
- C++ Programming Examples
- C++ Interview Questions and Answers (2025)

C++ Programming Language

-
-
-

C++ Programming Language

C++ is a programming language that is the foundation of many modern technologies like game

engines, web browsers, operating systems financial systems, etc. Bjarne Stroustrup developed it as an extension of the C language. C++ is generally used to create high-performance applications and provides better control of memory and system resources. The latest version of C++ is C++ 23 which was released in 2023.

Why Learn C++?

- C++ is one of the most used and popular programming languages. C++ is used in making operating systems, embedded systems, and Graphical User Interfaces.
- It supports object-oriented programming and implements all the OOP concepts such as Abstraction, Encapsulation, and Inheritance, which gives a clear structure to programs and allows code to be reused, lowering development costs and providing security.
- It makes programming easy for programmers to switch to C++ because its syntax is similar to C, Java, and C#.
- C++ is faster than Python and Java.
- It is useful in limited resource environments where C is also an option. It is also used where we need speed of execution or we need to work close to hardware.
- When compared to C, C++ has richer libraries, supports object oriented programming, templates, exception handling and many more features.

C++ Overview

- Introduction to C++
- Features of C++
- History of C++
- Interesting Facts about C++
- Setting up C++ Development Environment
- Similarities and Differences between C++ and C

C++ Basics

- First C++ Program
- C++ Basic Syntax
- C++ Comments
- C++ Tokens
- C++ Identifiers
- C++ Keywords
- Difference between Keyword and Identifier

C++ Variables and Constants

- C++ Variables
- C++ Constants
- Scope of C++ Variables
- C++ Storage Classes
- C++ Static Variables

C++ Data Types and Literals

- C++ Data Types
- C++ Literals
- C++ Derived Data Types
- C++ User-Defined Data Types
- C++ Data Type Ranges and Their Macros
- C++ Type Modifiers
- C++ Data Type Conversion
- C++ Typecasting Operators

C++ Operators

- C++ Operators
- C++ Arithmetic Operators

- C++ Unary Operators
- C++ Bitwise Operators
- C++ Relational Operators
- C++ Logical Operators
- C++ Assignment Operators
- C++ Ternary/Conditional Operators
- C++ Sizeof Operator
- C++ Scope Resolution Operator

C++ Input/Output

- C++ Basic Input / Output
- C++ Standard Input Stream (cin)
- C++ Standard Output Stream (cout)
- C++ Standard Error Stream (cerr)
- C++ Input / Output Manipulator

C++ Control Statements

- C++ Decision Making
- C++ if Statement
- C++ if-else Statement
- C++ if-else-if Ladder
- C++ Nested if-else Statement
- C++ Switch Statement
- C++ Jump Statements
- C++ Loops
- C++ for Loop
- C++ Range-Based for Loop
- C++ while Loop

- C++ do?while Loop

C++ Functions

- C++ Functions
- C++ return
- C++ Parameter Passing Techniques
- Call by Value vs Call by Reference
- C++ Default Arguments
- C++ Recursion
- C++ Inline Functions
- C++ Lambda Expression

C++ Pointers and References

- C++ Pointers and References
- C++ Pointers
- C++ Pointer Arithmetic
- Dangling, Void, Null, and Wild Pointers
- Applications of Pointers
- C++ nullptr
- C++ References
- Can references refer to an invalid location in C++?
- Difference Between Pointers and References in C++
- Passing by pointer Vs Passing by Reference in C++
- When do we pass arguments by reference or pointer?

C++ Arrays

- C++ Arrays
- C++ Multidimensional Arrays

- C++ Pointer to an Array
- Size of Array parameter
- Passing Arrays to Functions in C++
- What is Array Decay in C++? How can it be prevented?

C++ Strings

- C++ Strings
- C++ std::string Class
- C++ Array of Strings
- C++ String Functions
- C++ String Concatenation
- Tokenizing a String in C++
- C++ Substring

C++ Structures and Unions

- C++ Structures, Unions, and Enumerations
- C++ Structures
- C++ Pointer to Structure
- C++ Self-Referential Structures
- Difference Between C Structures and C++ Structures
- C++ Unions
- C++ Bit Fields
- C++ Enumeration
- C++ typedef
- Array of Structures vs Array within a Structure in C/C++

C++ Dynamic Memory Management

- C++ Dynamic Memory Management

- C++ new and delete Operators
- new vs malloc() and free() vs delete in C++
- Memory leak in C++
- Difference between Static and Dynamic Memory Allocation in C++

C++ Object-Oriented Programming

- C++ Object Oriented Programming (OOPs)
- C++ Classes and Objects
- C++ Access Modifiers
- C++ Friend Class and Function
- C++ Constructors
- C++ Default Constructors
- C++ Copy Constructor
- C++ Destructors
- C++ Private Destructor
- When is the Copy Constructor Called?
- Shallow Copy and Deep Copy in C++
- When Should We Write Our Own Copy Constructor?
- Does the Compiler Create a Default Constructor When We Write Our Own?
- C++ Static Data Members
- C++ Static Member Functions
- C++ this pointer
- C++ Scope Resolution Operator vs this pointer
- C++ Local Class
- C++ Nested Classes
- C++ enum Class
- Difference between Structure and Class in C++

- Why C++ is a partially Object Oriented Language?

C++ Encapsulation and Abstraction

- C++ Encapsulation
- C++ Abstraction
- Difference between Abstraction and Encapsulation in C++

C++ Polymorphism

- C++ Polymorphism
- C++ Function Overriding
- C++ Virtual Functions and Runtime Polymorphism
- Difference between Compile-time and Run-time Polymorphism in C++
- Difference between Inheritance and Polymorphism in C++

C++ Function Overloading

- C++ Function Overloading
- C++ Constructor Overloading
- C++ Functions that Cannot be Overloaded
- C++ Function Overloading and const Keyword
- C++ Function Overloading and Return Type
- C++ Function Overloading and float Data Type
- C++ Function Overloading and Default Arguments
- Can main() be overloaded?
- C++ Function Overloading Vs Function Overriding
- Advantages and Disadvantages of C++ Function Overloading

C++ Operator Overloading

- C++ Operator Overloading
- Types of C++ Operator Overloading

- C++ Functors
- C++ Operators that Cannot be Overloaded

C++ Inheritance

- C++ Inheritance
- C++ Inheritance Access
- C++ Multiple Inheritance
- C++ Hierarchical Inheritance
- C++ Multilevel Inheritance
- C++ Constructor in Multiple Inheritance
- C++ Inheritance and Friendship
- Does Function Overloading Work with Inheritance in C++?
- Difference Between Inheritance and Polymorphism in C++

C++ Virtual Functions

- C++ Virtual Functions
- C++ Virtual Functions in Derived Classes
- C++ Default Arguments and Virtual Function
- C++ Inline Virtual Functions
- C++ Virtual Destructor
- C++ Virtual Constructor
- C++ Virtual Copy Constructor
- C++ Pure Virtual Functions and Abstract Class
- C++ Pure Virtual Destructor in C++
- Can Static Functions be Virtual in C++?
- C++ RTTI (Run-Time Type Information)
- Can C++ Virtual Functions be Private?

C++ Exception Handling

- C++ Exception Handling
- C++ Exception Handling using Classes
- C++ Stack Unwinding
- C++ User-Defined Exceptions

C++ Files and Streams

- C++ Files and Streams
- C++ I/O Redirection

C++ Templates

- C++ Templates
- C++ Template Specialization
- C++ using Keyword

C++ Standard Template Library (STL)

- The C++ Standard Template Library (STL)
- STL Algorithms
- STL Containers
- STL Vector
- STL Pair
- STL Set
- STL Multiset
- STL Stack
- STL Queue
- STL Priority Queue
- STL Deque
- STL List

- STL Forward List
- STL Map
- STL Multimap
- STL Bitset
- STL Unordered Sets
- STL Unordered Multiset
- STL Unordered Map
- STL Unordered Multimap

C++ Iterators

- Introduction to C++ Iterators
- C++ Input Iterators
- C++ Output Iterators
- C++ Forward Iterators
- C++ Bidirectional Iterators
- C++ Random Access Iterators
- C++ istream_iterator and ostream Iterator
- Difference between C++ Iterators and Pointers

C++ Preprocessors

- C++ Preprocessor
- C++ Preprocessor Directives
- C++ #include Directive
- C++ #define Directive
- C++ Conditional Preprocessors
- Difference between C++ Preprocessor Directives and Function Templates

C++ Namespace

- C++ Namespaces
- Extending C++ Namespace and Unnamed Namespace
- Accessing, Creating Header, Nesting, and Aliasing Namespace
- C++ Inline Namespaces

Advanced C++

- C++ Multithreading
- C++ Smart Pointers
- Differences between Different C++ Smart Pointers
- Type of ?this? Pointer in C++
- Delete ?this? Pointer in C++
- Passing C++ Function as a Parameter
- C++ Signal Handling
- C++ Generics

C vs C++

- Differences and Similarities between C++ and C
- Difference between C++ and Objective C
- C programs that won?t compile in C++
- Program that produces different results in C and C++
- Void * in C vs C++
- Type Difference of Character Literals in C vs C++
- Difference between Structures in C and Structures in C++
- Cin-Cout vs Scanf-Printf

C++ vs Java

- Differences and Similarities between C++ and Java
- Inheritance in C++ vs Java

- Static keyword in C++ vs Java
- Default Virtual Behavior in C++ vs Java
- Exception Handling in C++ vs Java
- Foreach loop in C++ vs Java
- Templates in C++ vs Generics in Java
- Floating Point Operations & Associativity in C, C++, and Java

Competitive Programming in C++

- Competitive Programming ? A Complete Guide
- C++ Tricks for Competitive Programming
- Writing C/C++ code efficiently in Competitive Programming
- Why C++ is Best for Competitive Programming?
- Generating Test Cases in C++
- Fast I/O for Competitive Programming in C++
- Setting up Sublime Text for C++ Competitive Programming Environment
- Setting up VS Code for C++ Competitive Programming Environment
- Which C++ libraries are useful for competitive programming?
- Common mistakes to be avoided in Competitive Programming in C++

C++ Interview Questions

- Top 50 C++ Interview Questions and Answers
- Top C++ STL Interview Questions and Answers
- 30 OOPs Interview Questions and Answers
- Top C++ Exception Handling Interview Questions and Answers

Applications of C++

Here are the uses of C++ with real-world applications:

1. Operating Systems

C++ is most widely used programming language and become an ideal choice for developing operating systems. Mac OS X has majority of parts written in C++ and Most of Microsoft's software like Windows, Microsoft Office, IDE Visual Studio, and Internet Explorer are also written in C++.

2. Games

C++ is widely used for game development, and companies often choose it as their first option for developing gaming systems. C++ is a compiled language, meaning the code is directly translated into machine code that the processor can execute. This provides fast rendering and quick response times, which are crucial for any game. Additionally, C++'s proximity to the machine allows it to manipulate resources effectively, enabling the creation of complex 3D games and multiplayer games. The Unreal Engine, a popular game engine, uses C++ for these reasons.

3. Web Browsers

Most of the browsers in Computers are developed in C++ for effecting goals and Mozilla Firefox is totally developed by C++ and Google Applications and software like Chrome and Google File System are partly written in C++.

4. Compilers

Compilers of many programming languages are designed in C and C++ and this is because they are moderately lower-level when compared to other higher-level programming languages and C/C++ are closer to the hardware.

5. Embedded Systems

Embedded systems that need the program closer to the hardware such as smartwatches, medical equipment systems, mobile phones etc., are developed in C++ and It can perform a lot of low-level function calls, unlike different high-level programming languages.

5. Embedded Systems

- Recent Articles on C++

- C++ Programs
- C++ Interview Questions

1. What is the difference between C++ and C?

Answer:

1. What is the difference between C++ and C?

C++ | C

C++ was developed as an extension of C. | C is a developed independent language.

C++ supports OOPs concepts. | C does not support OOPs concepts.

Namespace is used by C++, to avoid name collisions. | The namespace feature is not supported by C.

1. What is the difference between C++ and C?

C++

1. What is the difference between C++ and C?

C

2. What is the difference between struct and class?

Answer:

2. What is the difference between struct and class?

C++ is a structure that is the same as a class except for a few differences like security. The difference between struct and class are given below:

2. What is the difference between struct and class?

Structure | Class

Members of the structure are public by default. | Members of the class are private by default.

When deriving a struct from a class/struct, default access specifiers for base class/struct are public. |

When deriving a class, default access specifiers are private.

Syntax:struct structure_name{type structure_member1;type structure_member2;}; | Syntax:class class_name{data_member;member_function;};

structure is generally used for objects with primary purpose of store data and have minimal or no methods | class is generally used for objects that represent entities with both data and behavior

2. What is the difference between struct and class?

Structure

2. What is the difference between struct and class?

Class

2. What is the difference between struct and class?

Syntax:struct structure_name{

2. What is the difference between struct and class?

type structure_member1;

2. What is the difference between struct and class?

type structure_member2;

2. What is the difference between struct and class?

};

2. What is the difference between struct and class?

Syntax:

2. What is the difference between struct and class?

class class_name{

2. What is the difference between struct and class?

data_member;

2. What is the difference between struct and class?

member_function;

2. What is the difference between struct and class?

};

2. What is the difference between struct and class?

structure is generally used for objects with primary purpose of store data and have minimal or no methods

2. What is the difference between struct and class?

class is generally used for objects that represent entities with both data and behavior

3. What do you mean by abstraction in C++?

Answer:

3. What do you mean by abstraction in C++?

Abstraction is the process of showing the necessary information to the user and hiding the details which programmers don't want to show to the user or hiding the details which are irrelevant to a particular user.

4. What is C++ best for?

Answer:

4. What is C++ best for?

C++ is used in developing browsers, operating systems, and applications, as well as in-game programming, software engineering, data structures, etc.

4. What is C++ best for?

Master C++ with our comprehensive C++ Programming Course Online, perfect for anyone from beginners to advanced learners. Learn everything from the fundamentals to advanced C++ concepts, and apply your knowledge through practical exercises. Take the Three 90 Challenge! Finish 90% of the course in 90 days, and enjoy a 90% refund. Stay focused, track your progress, and challenge yourself to complete the course! Join today and begin your C++ programming journey!

4. What is C++ best for?

- C++
- Tutorials

4. What is C++ best for?

- CPP

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge

- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions

- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Basic of DBMS

- Introduction of DBMS (Database Management System)
- History of DBMS
- Advantages of Database Management System
- Disadvantages of DBMS
- Application of DBMS
- Need for DBMS
- DBMS Architecture 1-level, 2-Level, 3-Level
- Difference between File System and DBMS

Entity Relationship Model

- Introduction of ER Model
- Structural Constraints of Relationships in ER Model
- Difference between entity, entity set and entity type
- Difference between Strong and Weak Entity
- Generalization, Specialization and Aggregation in ER Model
- Recursive Relationships in ER diagrams

Relational Model

- Introduction of Relational Model and Codd Rules in DBMS
- Types of Keys in Relational Model (Candidate, Super, Primary, Alternate and Foreign)
- Anomalies in Relational Model
- Mapping from ER Model to Relational Model
- Strategies for Schema design in DBMS

Relational Algebra

- Introduction of Relational Algebra in DBMS
- Basic Operators in Relational Algebra

- Extended Operators in Relational Algebra
- SQL Joins (Inner, Left, Right and Full Join)
- Join operation Vs Nested query in DBMS
- Tuple Relational Calculus (TRC) in DBMS
- Domain Relational Calculus in DBMS

Functional Dependencies

- Functional Dependency and Attribute Closure
- Armstrong's Axioms in Functional Dependency in DBMS
- Equivalence of Functional Dependencies
- Canonical Cover of Functional Dependencies in DBMS

Normalisation

- Introduction of Database Normalization
- Normal Forms in DBMS
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Boyce-Codd Normal Form (BCNF)
- Introduction of 4th and 5th Normal Form in DBMS
- The Problem of Redundancy in Database
- Database Management System | Dependency Preserving Decomposition
- Lossless Decomposition in DBMS
- Lossless Join and Dependency Preserving Decomposition
- Denormalization in Databases

Transactions and Concurrency Control

- Concurrency Control in DBMS
- ACID Properties in DBMS

- Implementation of Locking in DBMS
- Lock Based Concurrency Control Protocol in DBMS
- Graph Based Concurrency Control Protocol in DBMS
- Two Phase Locking Protocol
- Multiple Granularity Locking in DBMS
- Polygraph to check View Serializability in DBMS
- Log based Recovery in DBMS
- Timestamp based Concurrency Control
- Dirty Read in SQL
- Types of Schedules in DBMS
- Conflict Serializability in DBMS
- Condition of schedules to be View-equivalent
- Recoverability in DBMS
- Precedence Graph for Testing Conflict Serializability in DBMS
- Database Recovery Techniques in DBMS
- Starvation in DBMS
- Deadlock in DBMS
- Types of Schedules based Recoverability in DBMS
- Why recovery is needed in DBMS

Indexing, B and B+ trees

- Indexing in Databases - Set 1
- Introduction of B-Tree
- Insert Operation in B-Tree
- Delete Operation in B-Tree
- Introduction of B+ Tree
- Bitmap Indexing in DBMS

- Inverted Index
- Difference between Inverted Index and Forward Index
- SQL Queries on Clustered and Non-Clustered Indexes

File organization

- File Organization in DBMS - Set 1
- File Organization in DBMS | Set 2
- File Organization in DBMS | Set 3

DBMS Interview questions and Last minute notes

- Last Minute Notes - DBMS
- Commonly asked DBMS interview questions
- Commonly asked DBMS Interview Questions | Set 2

DBMS GATE Previous Year Questions

- Database Management System - GATE CSE Previous Year Questions
- Database Management Systems | Set 2
- Database Management Systems | Set 3
- Database Management Systems | Set 4
- Database Management Systems | Set 5
- Database Management Systems | Set 6
- Database Management Systems | Set 7
- Database Management Systems | Set 8

DBMS Tutorial ? Learn Database Management System

-
-
-

DBMS Tutorial ? Learn Database Management System

Database Management System (DBMS) is a software used to manage data from a database.

DBMS Tutorial ? Learn Database Management System

- A database is a structured collection of data that is stored in an electronic device. The data can be text, video, image or any other format.
- A relational database stores data in the form of tables and a NoSQL database in the form of key-value pairs.
- SQL (Structured Query Language) is a standard language for accessing and manipulating data in a relational database.
- A DBMS is a software that allows to create, update and retrieval of data in an organized way. It also provides security to the database.
- Examples of relational DBMS are MySQL, Oracle, Microsoft SQL Server, PostgreSQL and Snowflake.
- Examples of NoSQL DBMS are MongoDB, Cassandra, DynamoDB and Redis.

DBMS Tutorial ? Learn Database Management System

DBMS Tutorial ? Learn Database Management System

In this Database Management System tutorial, you will learn basic to advanced topics like the ER model, Relational Model, Relation Algebra, Normalization, File Organization, etc.

Introduction

- Need For DBMS
- DBMS Introduction
- DBMS Architecture (1, 2 and 3 Tier)

Entity Relationship Model

- ER Model

- Enhanced ER Model
- Minimization of ER Diagram
- Generalization, Specialization and Aggregation
- Recursive Relationships

Relational Model and Functional Dependencies

- Relational Model
- Keys (Candidate, Super, Primary, Alternate and Foreign)
- Functional Dependency and Attribute Closure
- Finding Attribute Closure and Candidate Keys using Functional Dependencies
- Armstrong's Axioms in Functional Dependency
- Equivalence of Functional Dependencies
- Canonical Cover
- Anomalies in Relational Model
- Mapping from ER Model to Relational Model
- Strategies for Schema design
- Schema Integration

Relational Model and Functional Dependencies

>> Quiz on ER and Relational Model

Normalization

- Introduction
- Normal Forms
- Minimum relations satisfying 1NF
- The Problem of redundancy in Database
- Dependency Preserving Decomposition
- Lossless Join Decomposition

- Lossless Join and Dependency Preserving Decomposition
- How to find the Highest Normal Form of a Relation
- Introduction of 4th and 5th Normal form
- Domain Key normal form
- Denormalization in Databases
- DBMS | Data Replication

Normalization

>> Quiz on Normal Forms

Relational Algebra and Calculus

- Introduction
- Basic Operators
- Extended Operators
- Inner Join vs Outer Join
- Join operation Vs nested query
- Tuple Relational Calculus
- Row oriented vs. column oriented data stores
- How to solve Relational Algebra Problems for GATE

Transactions and Concurrency Control

- Introduction
- ACID Properties
- Concurrency Control -Introduction
- Types of Schedules
- Conflict Serializability
- View Serializability
- How to test if two schedules are View Equal or not ?

- Recoverability of Schedules
- Precedence Graph for testing Conflict Serializability
- Transaction Isolation Levels in DBMS
- Implementation of Locking in DBMS
- Lock Based Protocol
- Graph Based Protocol
- Two Phase Locking (2-PL)
- Multiple Granularity Locking
- Thomas Write Rule
- Polygraph to check View Serializability
- Log based recovery
- Timestamp Ordering Protocols
- TimeStamp and Deadlock Prevention Schemes
- Dirty read in SQL
- Database Recovery Techniques
- Starvation in DBMS
- Deadlock in DBMS
- DBMS | OLAP vs OLTP
- Types of OLAP Systems
- Types of Recoverability of Schedules and easiest way to test schedule | Set 2

Transactions and Concurrency Control

>> Quiz on Transactions and concurrency control

Indexing, B and B+ trees

- Indexing and its Types
- B-Tree | Set 1 (Introduction)
- B-Tree | Set 2 (Insert)

- B-Tree | Set 3 (Delete)
- B+ Tree (Introduction)
- Bitmap Indexing
- Inverted Index
- Difference between Inverted Index and Forward Index
- SQL queries on clustered and non-clustered Indexes

Indexing, B and B+ trees

>> Practice questions on B and B+ Trees>> Quizzes on Indexing, B and B+ Trees

File Organization

- File Organization ? Set 1
- File Organization ? Set 2 (Hashing in DBMS)
- File Organization ? Set 3
- File Organization ? Set 4

File Organization

>> Quiz on File structures

Advanced Topics

- Multimedia Database
- RAID
- Query Optimization
- How to store a password in database?
- Storage Area Networks
- Network attached storage
- ODBMS ? Definition and overview
- Difference between RDBMS and HBase
- Challenges of database security

- Federated database management system issues
- Distributed Database System
- Functions of Distributed Database System
- Semantic Heterogeneity
- Advantages of Distributed database
- Comparison ? Centralized, Decentralized and Distributed Systems
- Characteristics of Biological Data (Genome Data Management)
- Data Management issues in Mobile database
- Future Works in Geographic Information System
- Difference between Structured, Semi-structured and Unstructured data
- Web Information Retrieval | Vector Space Model

Data Warehouse and Data Mining

- Star Schema in Data Warehouse modeling
- Data Warehouse Modeling | Snowflake Schema
- Dimensional Data Modeling
- Data Warehousing
- Data Warehouse Architecture
- Characteristics and Functions of Data warehouse
- Difficulties of Implementing Data Warehouses
- Data Mining
- Data Mining | KDD process
- Data Mining | Sources of Data that can be mined
- Data Marts
- Architecture of HBase
- Apache HBase
- Architecture and Working of Hive

- Apache Hive
- Difference between Hive and HBase

Interview Questions

- DBMS interview questions | Set 1
- DBMS interview questions | Set 2
- SQL Interview Questions
- SQL Query Interview Questions
- PostgreSQL Interview Questions
- PL/SQL Interview Questions

SQL Tutorial

- SQL | Tutorial
- Quiz on SQL

Misc Topics

- DBMS Interfaces
- Categories of DBMS Users
- Data Abstraction and Data Independence
- Database Objects
- Impedance Mismatch

DBMS practices questions

- Database Management Systems | Set 1
- Database Management Systems | Set 2
- Database Management Systems | Set 3
- Database Management Systems | Set 4
- Database Management Systems | Set 5
- Database Management Systems | Set 6

- Database Management Systems | Set 7
- Database Management Systems | Set 8
- Database Management Systems | Set 9
- Database Management Systems | Set 10
- Database Management Systems | Set 11

Advantages of DBMS

There are some following reasons to learn DBMS:

Advantages of DBMS

- Organizing and management of data:DBMS helps in managing large amounts of data in an organized manner. It provides features like create, edit, delete, and read.
- Data Security:DBMS provides Security to the data from the unauthorized person.
- Improved decision-making: From stored data in the database we can generate graphs, reports, and many visualizations which helps in decision-making.
- Consistency:In a traditional database model all things are manual or inconsistent, but DBMS enables to automation of the operations by queries.

Disadvantages of DBMS

- Complexity:DBMS can be hard to design, implement, and manage, needing specialized knowledge.
- Cost:High setup costs, including hardware, software, and skilled personnel, can be expensive. Ongoing maintenance adds to the cost.
- Performance Overhead:DBMS might slow down simple tasks due to their extra features and general-purpose nature.
- Security Risks:Centralizing data can create security risks. If the system is hacked, all data could be compromised.
- Resource Intensive:DBMS need a lot of memory, storage, and processing power, which can be

costly.

- Data Integrity Issues: Complex systems can lead to data integrity problems if not managed well.

Conclusion

Understanding Database Management Systems (DBMS) is essential for managing and organizing data effectively. This DBMS tutorial has introduced you to key concepts like database models, SQL queries, normalization, and data security. With this knowledge, you can design efficient databases, maintain data integrity, and improve performance.

Quick Links:

- Last Minutes Notes(LMNs) on DBMS
- Quizzes on DBMS
- Practice Problems on DBMS

What is Database?

A database is a collection of organized data which can easily be created, updated, accessed, and managed. Records are kept maintained in tables or objects. A tuple (row) represents a single entry in a table. DBMS manipulates data from the database in the form of queries given by the user.

What are different languages present in DBMS?

- DDL (Data Definition Language): These are the collection of commands which are required to define the database.E.g., CREATE, ALTER, RENAME, TRUNCATE, DROP, etc.
- DML (Data Manipulation Language): These are the collection of commands which are required to manipulate the data stored in a database.E.g., SELECT, UPDATE, INSERT, DELETE, etc.
- DCL (Data Control Language): These are the collection of commands which are dealt with the user permissions and controls of the database system.E.g, GRANT, and REVOKE.
- TCL (Transaction Control Language): These are the collection of commands which are required to deal with the transaction of the database.E.g., COMMIT, ROLLBACK, and SAVEPOINT.

What are the ACID properties in DBMS?

The full form of ACID is Atomicity, Consistency, Isolation, and Durability these are the properties of DBMS that ensure a safe and secure way of sharing data among multiple users. A? Atomic: All changes to the data must be performed successfully or not at all. C? Consistent: Data must be in a consistent state before and after the transaction. I? Isolated: No other process can change the data while the transaction is going on. D? Durable: The changes made by a transaction must persist.

What are the Advantages of DBMS?

The followings are the few advantages of DBMS :

What are the Advantages of DBMS?

- Data Sharing: Data from the same database can be shared by multiple users at the same time.
- Integrity: It allows the data stored in an organized and refined manner.
- Data Independence: It allows changing the data structure without changing the composition of executing programs.
- Data Security: DBMS comes with the tools to make the storage and transfer of databases secure and reliable. Authentication and encryption are the tools used in DBMS for data security.

What are the Advantages of DBMS?

Dreaming of M.Tech in IIT? Get AIR under 100 with our GATE 2026 CSE & DA courses! Get flexible weekday/weekend options, live mentorship, and mock tests. Access exclusive features like All India Mock Tests, and Doubt Solving? your GATE success starts now!

What are the Advantages of DBMS?

- GATE CS
- DBMS
- dbms

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++

- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System

- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics

- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates

- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD

- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course

- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction

- What is DevOps ?
- DevOps Lifecycle
- The Evolution of DevOps - 3 Major Trends for Future

Version Control

- Version Control Systems
- Merge Strategies in Git
- Which Version Control System Should I Choose?

Continuous Integration (CI) & Continuous Deployment (CD)

- What is CI/CD?
- 15 Best CI/CD Tools That You Should Know
- Understanding Deployment Automation

Containerization

- What is Docker?
- What is Dockerfile Syntax?
- Kubernetes - Introduction to Container Orchestration

Orchestration

- Kubernetes - Introduction to Container Orchestration
- Fundamental Kubernetes Components and their role in Container Orchestration
- How to Use AWS ECS to Deploy and Manage Containerized Applications?

Infrastructure as Code (IaC)

- What is Infrastructure as Code (IaC)?
- Introduction to Terraform
- What is AWS Cloudformation?

Monitoring and Logging

- Working with Prometheus and Grafana Using Helm
- Working with Monitoring and Logging Services

Security in DevOps

- What is DevSecOps: Overview and Tools
- DevOps Best Practices for Kubernetes

DevOps Tutorial

-
-
-

DevOps Tutorial

DevOps is a collection of two words, "Development" and "Operations," representing a cultural approach that emphasizes collaboration between development and operations teams to streamline the entire software delivery lifecycle.

DevOps Tutorial

This DevOps tutorial is designed to help learn DevOps basics and advanced concepts, including Git, Ansible, Docker, Puppet, Jenkins, a range of DevOps tools, Azure DevOps, Chef, Nagios, and Kubernetes.

DevOps Tutorial

DevOps Tutorial

DevOps Tutorial

Table of Content

DevOps Tutorial

- What is DevOps?
- 1. DevOps Fundamentals
- Linux
- Source Code Management
- YAML
- Cloud
- Docker
- Kubernetes
- Infrastructure as a Code
- DevOps Courses
- Why Learn DevOps?
- DevOps Career Opportunities
- Is DevOps for Freshers?

What is DevOps?

DevOps is a software development approach emphasizing collaboration, automation, and continuous delivery to provide high-quality products to customers quickly and efficiently. DevOps breaks down silos between development and operations teams to enable seamless communication, faster time-to-market, and improved customer satisfaction.

What is DevOps?

It allows a team to handle the complete application lifecycle, from development to testing, operations, and deployment. It shows cooperation between Development and Operations groups to deploy code to production quickly in an automated and repeatable manner.

What is DevOps?

Every phase of the software development lifecycle, including planning, coding, testing, deployment, and monitoring, is heavily automated in DevOps. This improves productivity, ensures consistency,

and lowers error rates in the development process. A culture of continuous improvement is also promoted by DevOps, where feedback loops are incorporated into the procedure to facilitate quicker iteration and better decision-making. Organizations can increase their agility, lower costs, and speed up innovation by adopting DevOps.

1. DevOps Fundamentals

In this free DevOps Tutorial we are going to discuss the proper curriculum that you need to cover to become a DevOps Engineer. While getting started with DevOps, you first need to understand the fundamentals of DevOps like what is DevOps, how it works, etc.

1. DevOps Fundamentals

- Introduction to DevOps
- Evolution of DevOps
- How DevOps Works?
- Lifecycle of DevOps
- DevOps Pipeline and Methodology
- Agile vs DevOps
- Skills and Salary for a DevOps Engineer
- How to Become a DevOps Engineer
- DevOps Best Practices
- 30 Best DevOps Tools to Master in 2023
- DevOps Tools to Try in 2022
- 6 DevOps Certifications in 2021
- 10 DevOps Skills in 2021
- Difference between Cloud and DevOps Engineer

2. Linux

Linux is one of the most popular operating systems for servers and cloud-based infrastructures. It

gives access to a robust CLI, a scripting environment, essential tools and utilities, strong security features, and powerful diagnostic tools for troubleshooting. In order to master the art of delivering high-quality software and infrastructure, it is required for a DevOps Engineer to master Linux.

2. Linux

- Linux Commands
- Introduction to Linux Shell Scripting
- How to create a Shell Script
- Introduction to Bash and Bash Scripting
- Debian Software Package Management (dpkg) in Linux
- Linux Networking Tools
- Network configuration and troubleshooting commands in Linux
- Top 10 Linux Server Security Tips
- SSH Server (sshd) Configuration and Security Options With Examples
- Linux Virtualization: Linux Containers
- How to Manage Linux Containers using LXC

3. Source Code Management

Source Code Management is one of the key aspects of DevOps. It is the practice of tracking and managing the versions of your source code. Git is considered to be one of the best tools for version control of source codes. It allows DevOps Engineers to collaborate, manage code, and implement CI/CD pipelines, code quality, and Infrastructure as a Code.

3. Source Code Management

- Introduction and Installation of Git
- How Git Version Control Works?
- Useful Git Commands and Basic Concepts
- Introduction to GitHub

- List of useful GitHub Commands
- Difference Between GitLab and GitHub
- How to setup GitLab Repo in Windows 10
- How to download and install GitLab?
- Bitbucket vs GitHub vs GitLab

4. YAML

Yet Another Markup Language or YAML Ain't Markup Language is a popular data serialization language that is used extensively in DevOps for defining and managing Infrastructure as a Code, managing configuration for various tools, defining CI/CD Pipelines and integrating with other technologies like JSON, XML, and Python, which makes it easy to learn and use in different DevOps scenarios.

4. YAML

- YAML Comments
- How to block comments in YAML
- Difference between YAML and JSON

5. Cloud

Cloud Computing is an essential tool to learn to become a DevOps Engineer. Many modern days software applications are deployed on Cloud Platforms like Amazon Web Service, Microsoft Azure, Google Cloud Platform, etc as it provides scalability of the resources, elasticity, automation, tools integrations, and cost-optimizations which helps to manage software applications in the cloud.

5. Cloud

- What is AWS
- How to setup AWS
- How to create an Instance in EC2
- What is a Load Balancer

- How to create a Load Balancer?
- Scaling in EC2
- Create AutoScaling Group in EC2
- What is DNS
- What is Route53
- AWS VPC
- Introduction to Microsoft Azure
- Data Security in Azure
- Virtual Network in Azure
- Management of Virtual Machines in Azure
- Introduction to Google Cloud Platform
- Google Cloud Platform Networking Services

6. Docker

Docker is a popular containerization tool that is used to deliver software quickly by using the concept of containerized code which helps for easy management and maintenance of applications.

6. Docker

- Introduction to Docker
- Docker Architecture
- Docker Instructions
- Dockerfile
- Docker Images
- Introduction to Docker Compose
- Docker Storage
- Docker Networking
- Docker Ports
- Docker Registry

- Introduction to Docker Swarm

7. Kubernetes

Kubernetes is a popular container orchestration tool that provides application scalability, resilience, flexibility, portability, automation, monitoring, and access to a rich ecosystem of tools and plugins which can be applied to various aspects of the DevOps lifecycle, such as deployment, scaling, monitoring, and troubleshooting of applications.

7. Kubernetes

- Introduction to Kubernetes
- Kubernetes ? Architecture
- Kubernetes ? Deployments
- Kubernetes ? Volumes
- Kubernetes ? Secrets
- Kubernetes ? Kubectl
- Kubernetes ? ConfigMap

8. Infrastructure as a Code

IaC enables automating and configuring the infrastructure resources using various tools such as Terraform, CloudFormation, ARM Templates, etc. It defines cloud resources, manages resource dependencies, creates reusable templates, tests IaC code, and manages code changes using version control systems.

8. Infrastructure as a Code

- Introduction to Terraform
- Introduction to Gradle
- Introduction to Maven
- How to Test Java Applications Using TestNG?
- How to Integrate JMeter with Prometheus and Grafana?

- How to Run Only One Unit Test Class Using Gradle?
- Automation using Chef
- Using Ansible to Manage Remote Machines
- Introduction to Jenkins
- How to Make a CI-CD Pipeline in Jenkins?
- Introduction to Puppet
- Introduction to Docker Swarm
- Microsoft Azure ? Deploying Web Applications Using GitHub Actions
- Microsoft Azure ? Build and Deploy App with Azure SDK For Java
- What is AWS OpsWorks?
- Microsoft Azure ? Configuring Backup For Azure App Services
- Microsoft Azure ? Configuring an Alert Rule with Azure Monitor
- Microsoft Azure ? Configuring Backup For Azure App Services
- Microsoft Azure ? Configure Azure SQL with Azure CLI
- How to Integrate JMeter with Prometheus and Grafana?
- Working with Prometheus and Grafana Using Helm

DevOps Courses

- DevOps Bootcamp ? Self-Paced Course
- DevOps Engineering ? Planning to Production

DevOps Courses

Also Read:

DevOps Courses

- How to Become a DevOps Engineer?
- Complete DevOps Roadmap ? Beginner to Advanced

Why Learn DevOps?

Some key features of DevOps are given below:

Why Learn DevOps?

- The operation and development teams cooperate together.
- Testing and deployment are performed respectively after the design-build.
- Manual code deployment often leads to human errors in production, and DevOps helps to automate it.

DevOps Career Opportunities

Once you got a mastery in DevOps, then there are multiple career open in front of you. Here in this section we have listed best career opportunities.

DevOps Career Opportunities

- DevOps Engineer
- Site Reliability Engineer
- Cloud DevOps Engineer
- Automation Engineer
- Security DevOps Engineer
- Release Engineer
- CI/CD Engineer
- DevOps Consultant
- Toolchain Developer
- Quality Assurance (QA) Engineer with DevOps Skills

List of Companies Using DevOps

Company Name | DevOps Implementation

Amazon | Extensive use of DevOps in AWS, CI/CD pipelines, and automation tools.

Google | DevOps practices for managing large-scale infrastructure, automation, and microservices.

Netflix | DevOps for continuous delivery, auto-scaling, and rapid deployment of features.

Facebook (Meta) | DevOps for managing large-scale data centers, automation, and site reliability.

Microsoft | DevOps integration in Azure, automated deployments, and CI/CD practices.

Etsy | Early adopter of DevOps, using it for rapid deployment and continuous integration.

Uber | Utilizes DevOps for microservices, automation, and scaling infrastructure.

Airbnb | Implements DevOps for infrastructure as code, automation, and scaling operations.

Spotify | DevOps for continuous integration, automation, and microservices architecture.

LinkedIn | Uses DevOps for infrastructure management, automation, and rapid feature releases.

Is DevOps for Freshers?

DevOps is a hot topic in the IT industry and lots of companies now need a DevOps Engineer to manage their servers, code deployment process, and maintenance of their applications. If you also want to join any organization as a DevOps Engineer without any prior work experience, then it is very important for you to follow these certain tips to get into the world of DevOps.

Is DevOps for Freshers?

- Learn the Fundamentals listed above
- Gain Hands-on knowledge by practicing and building projects
- Try to learn and master automation
- Develop soft skills
- Network with professionals
- Always be in the loop of learning and implementing

Conclusion

DevOps is an excellent approach for delivering quality products more efficiently and helps an organization to stay competitive in today's fast-paced digital world by focusing more on automating services, collaborations, and continuous improvement of the product's features by adapting continuous integration and continuous delivery methodology.

What is DevOps?

DevOps is a methodology that promotes collaboration between development and operations teams, aiming to streamline software development, delivery, and deployment processes.

What are the key benefits of implementing DevOps?

DevOps offers several benefits, including faster software delivery, improved collaboration and communication, enhanced quality and reliability, increased efficiency and productivity, and better customer satisfaction.

Which tools are commonly used in DevOps?

Popular DevOps tools include Git, Jenkins, Ansible, Docker, Puppet, Chef, Nagios, Kubernetes, and Azure DevOps. These tools assist in various aspects of the software development lifecycle, such as version control, continuous integration, configuration management, and containerization.

Is DevOps a coding?

No, DevOps is not coding. DevOps is a methodology and cultural approach that emphasizes collaboration and integration between development and operations teams to streamline software delivery processes. While coding is an important aspect of software development, DevOps encompasses a broader set of practices, tools, and principles beyond coding.

Is DevOps a coding?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

Is DevOps a coding?

- DevOps

- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy

- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Django REST API ? CRUD with DRF

-
-
-

Django REST API ? CRUD with DRF

Django REST Framework is used to create web APIs very easily and efficiently. This is a wrapper around the Django Framework. There are three stages before creating an API through the REST framework, Converting a Model's data to JSON/XML format (Serialization), Rendering this data to the view, and Creating a URL for mapping to the views.

Django REST API ? CRUD with DRF

For this tutorial, we will be building simple REST APIs for supermarket applications. This store contains a variety of food, beverages, and household products, organized into various categories. These categories have subcategories and subcategories have items under them. Each item has a price for which it can be sold. Now we should be able to add, update, view, and delete such records from our database using DRF APIs.

Django REST API ? CRUD with DRF

Let's assume you have Django set up and installed in your system. In case you don't have any set up at all then refer to the below articles ?

Django REST API ? CRUD with DRF

- Django Introduction and Installation
- How to Create a Basic Project using MVT in Django?
- How to Create an App in Django?

Install Django REST Framework

To install Django REST Framework on Windows, Linux or any operating system, one would need

pip package. To check how to install pip on your operating system, check out ? PIP Installation ?Windows||Linux.

Install Django REST Framework

Now, run a simple command,

Install Django REST Framework

Install Django REST Framework

After installing the REST framework, go to settings.py, and in INSTALLED_APPS add ?rest_framework? at the bottom.

Python3

Python3

Python3

Python3

```
INSTALLED_APPS=['django.contrib.admin','django.contrib.auth','django.contrib.contenttypes','django.contrib.sessions','django.contrib.messages','django.contrib.staticfiles','rest_framework',]
```

Python3

For more information refer to our article [Django REST Framework Installation](#).

Creating App for Django REST Framework

After installing the DRF and adding it to settings.py, let's create an app using the command ?

Creating App for Django REST Framework

A folder with the name api would have been registered by now.

Creating App for Django REST Framework

Creating App for Django REST Framework

Let's add this app to `INSTALLED_APPS` and `urls.py` also.

Creating App for Django REST Framework

In, `settings.py`,

Python3

Python3

Python3

Python3

```
INSTALLED_APPS=[  
'django.contrib.admin',  
'django.contrib.auth',  
'django.contrib.contenttypes',  
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
'rest_framework',  
'api.apps.ApiConfig',  
,
```

Python3

Now, add api urls in `urls.py`. In `ngfg_shopping.urls.py`,

Python3

Python3

Python3

Python3

```
from django.contrib import admin  
from django.urls import path, include  
urlpatterns = [path('admin/',  
admin.site.urls), path('api/', include('api.urls'))]
```

Creating Model in Django

Now let's create our model. We will create an item model. This model will be used by API to perform

the CRUD operations.

Python3

Python3

Python3

Python3

```
from django.db import models
class Item(models.Model):
    category = models.CharField(max_length=255)
    subcategory = models.CharField(max_length=255)
    name = models.CharField(max_length=255)
    amount = models.PositiveIntegerField()
    def __str__(self):
        return self.name
```

Python3

Now after our app gets ready let's create the serializer for our Item class.

Serializing Django Objects

Serializers in Django REST Framework convert the objects into data types that are understandable by javascript and front-end frameworks. Serializers also provide deserialization, allowing parsed data to be converted back into complex types, after first validating the incoming data. The two major serializers that are most popularly used are `ModelSerializer` and `HyperLinkedModelSerializer`.

Serializing Django Objects

For this tutorial, we will use the `ModelSerializer`. If you want to know more about creating serializers manually then refer to the below articles ?

Serializing Django Objects

- Serializers ? Django REST Framework
- `HyperlinkedModelSerializer` in serializers ? Django REST Framework

ModelSerializer

ModelSerializer is a layer of abstraction over the default serializer that allows to quickly create a serializer for a model in Django. It provides a shortcut that lets you automatically create a Serializer class with fields that correspond to the Model fields. The ModelSerializer class is the same as a regular Serializer class, except that:

ModelSerializer

- It will automatically generate a set of fields for you, based on the model.
- It will automatically generate validators for the serializer, such as unique_together validators.
- It includes simple default implementations of .create() and .update().

ModelSerializer

Now let's create our serializers.py file in the api folder and add the below code ?

Python3

Python3

Python3

Python3

```
from django.db.models import fields
from rest_framework import serializers
from .models import Item

class ItemSerializer(serializers.ModelSerializer):
    class Meta:
        model = Item
        fields = ('category', 'subcategory', 'name', 'amount')
```

Create Views for Django

To render data into frontend, and handle requests from user, we need to create a view. In Django REST Framework, we call these viewsets, so let's create a view in apis/views.py,

Python3

Python3

Python3

Python3

```
from rest_framework.decorators import api_view
from rest_framework.response import Response
from models import Item
from serializers import ItemSerializer

@api_view(['GET'])
def api_overview(request):
    api_urls = {
        'all_items': '/',
        'Search by Category': '/?category=category_name',
        'Search by Subcategory': '/?subcategory=category_name',
        'Add': '/create',
        'Update': '/update/pk',
        'Delete': '/item/pk/delete'
    }
    return Response(api_urls)
```

Python3

In the above code, the `api_view` decorator takes a list of HTTP methods that a view should respond to. Other methods will respond with the Method Not Allowed.

Python3

Now let's update our `api/urls.py` file ?

Python3

Python3

Python3

Python3

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.ApiOverview, name='home')]

```

Python3

Now let's run our server. Run the following commands ?

Python3

Now head to `http://127.0.0.1:8000/api/`

Python3

Python3

Now let's implement our CRUD Operations through our API.

Django Rest Framework ? Create View

Now our create view will use the POST method for inserting data into our database. Let's create our `add_items` function in the `views.py` file.

Python3

Python3

Python3

Python3

```
from rest_framework import serializers
from rest_framework import status
@api_view(['POST'])
def add_items(request):
    item = ItemSerializer(data=request.data)
    # validating for already existing data
    if Item.objects.filter(**request.data).exists():
        raise serializers.ValidationError('This data already exists')
    if item.is_valid():
        item.save()
        return Response(item.data)
    else:
        return Response(status=status.HTTP_404_NOT_FOUND)
```

Python3

Now let's update our `urls.py` file and add the endpoint for the create view function we just created.

Python3

Python3

Python3

Python3

```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.ApiOverview, name='home'),
    path('create/', views.add_items, name='add-items'),
]
```

Python3

Visit <http://127.0.0.1:8000/api/create/>

Python3

Django Rest Framework ? Read View

Now our list view will use the GET method for retrieving data from our database. Let's create our view_items function in the views.py file. This view_items function will either show all the data or filtered data queried by the user according to the category, subcategory, or name.

Django Rest Framework ? Read View

In views.py

Python3

Python3

Python3

Python3

```
@api_view(['GET'])def view_items(request):#    checking    for    the    parameters    from    the    URLLif request.query_params:items=Item.objects.filter(**request.query_params.dict())else:items=Item.objects.all()# if there is something in items else raise errorif items:serializer=ItemSerializer(items,many=True)return Response(serializer.data)else:return Response(status=status.HTTP_404_NOT_FOUND)
```

Python3

In urls.py

Python3

Python3

Python3

Python3

```
from django.urls import path from . import views urlpatterns = [path("", views.ApiOverview,
name='home'), path('create/', views.add_items, name='add-items'), path('all/', views.view_items,
name='view_items'),]
```

Python3

Now visit <http://127.0.0.1:8000/api/all/>

Python3

Python3

This view function also lets us filter by category or subcategory. You can use either of the following URLs i.e. http://127.0.0.1:8000/api/?category=category_name or http://127.0.0.1:8000/api/?subcategory=category_name to filter for both category and subcategory respectively. You can also use http://127.0.0.1:8000/api/all/?name=item_name to search for a specific item.

Python3

If we visit <http://127.0.0.1:8000/api/all/?category=food> our search result will narrow down to ?

Python3

Django Rest Framework ? Update View

Now for our update view function we will use the POST method. Let's create our `update_items` function in the `views.py` file. This view function will update a particular item from the database. It will filter the item with the help of the primary key.

Django Rest Framework ? Update View

In `views.py`

Python3

Python3

Python3

Python3

```
@api_view(['POST'])defupdate_items(request, pk):item=Item.objects.get(pk=pk)data=ItemSerializer(instance=item, data=request.data)ifdata.is_valid():data.save()returnResponse(data.data)else:returnResponse(status=status.HTTP_404_NOT_FOUND)
```

Python3

In urls.py

Python3

Python3

Python3

Python3

```
fromdjango.urlsimportpathfrom.importviewsurlpatterns=[path("", views.ApiOverview, name='home'),path('create/', views.add_items, name='add-items'),path('all/', views.view_items, name='view_items'),path('update/<int:pk>/', views.update_items, name='update-items'),]
```

Python3

Now head to <http://127.0.0.1:8000/api/all/?name=potato>

Python3

Django Rest Framework ? Delete View

For our delete view function we will use the DELETE method. Let's create our delete_items function in the views.py file. This view function will delete a particular item from the database.

Django Rest Framework ? Delete View

In views.py

Python3

Python3

Python3

Python3

```
@api_view(['DELETE'])defdelete_items(request, pk):item=get_object_or_404(Item, pk=pk)item.delete()returnResponse(status=status.HTTP_202_ACCEPTED)
```

Python3

In urls.py

Python3

Python3

Python3

Python3

```
fromdjango.urlsimportpathfrom.importviewsurlpatterns=[path("", views.ApiOverview, name='home'),path('create/', views.add_items, name='add-items'),path('all/', views.view_items, name='view_items'),path('update/<int:pk>/', views.update_items, name='update-items'),path('item/<int:pk>/delete/', views.delete_items, name='delete-items'),]
```

Python3

Now visit <http://127.0.0.1:8000/api/item/pk/delete/>. See the below GIF for better understanding.

Python3

Python3

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through hands-on projects and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

Python3

- Python
- Django-REST
- Python Django

Python3

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy

- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Django Basics

- Django Basics
- Django Installation and Setup
- When to Use Django? Comparison with other Development Stacks
- Django Project MVT Structure
- How to Create a Basic Project using MVT in Django ?
- How to Create an App in Django ?
- Django settings file - step by step Explanation

Django view

- Views In Django | Python
- Django Function Based Views
- Django Class Based Views
- Class Based vs Function Based Views - Which One is Better to Use in Django?

Django Model

- Django Models
- Django model data types and fields list
- Built-in Field Validations - Django Models
- How to use User model in Django?
- Meta Class in Models - Django
- get_object_or_404 method in Django Models

Django Forms

- Django Forms
- How to create a form using Django Forms ?
- Django Form | Data Types and Fields
- Django Form | Build in Fields Argument

- Python | Form validation using django
- Render Django Form Fields Manually

Django URLs

- Django URL patterns | Python
- Get parameters passed by urls in Django
- url - Django Template Tag
- URLField - Django Models
- URL fields in serializers - Django REST Framework

More topics on Django

- Handling Ajax request in Django
- Python | User groups with Custom permissions in Django
- Python | Django Admin Interface
- Python | Extending and customizing django-allauth
- Django - Dealing with warnings
- Python | Sessions framework using django
- Django Sign Up and login with confirmation Email | Python

Projects

- Python Django | Google authentication and Fetching mails from scratch
- ToDo webapp using Django
- Python | Django News App
- Weather app using Django | Python
- College Management System using Django - Python Project
- E-commerce Website using Django
- Create Word Counter app using Django
- Youtube video downloader using Django

- Voting System Project Using Django Framework
- Project Idea - A website acting as transaction between oxygen sellers and buyers

Django Interview Questions and Answers

- Top 50 Django Interview Questions and Answers
- Django Interview Questions & Answers With Practical Tips For Junior Developers

Django Tutorial | Learn Django Framework

-
-
-

Django Tutorial | Learn Django Framework

Django, built with Python, is designed to help developers build secure, scalable, and feature-rich web applications quickly and efficiently. Whether you're a beginner looking to create your first dynamic website or an experienced developer aiming to enhance your skills, this tutorial will guide you through Django's core concepts.

Django Tutorial | Learn Django Framework

This also tutorial provides you with Django projects to help you apply your knowledge and build some cool web applications. These projects not only provide you with experience in building with the Django framework but will also add value to your resume. This Python Django tutorial teaches basic to advanced Django concepts for backend development. Learn topics like forms, templates, views, ORM, etc.

Python Django

Python-based web framework Django allows you to create efficient web applications quickly. It is also called batteries included web framework Django because it provides built-in features for everything including Django Admin Interface, default database ? SQLite3, etc.

Python Django

Python Django

When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django Python gives you ready-made components to use for rapid development. There are many more benefits of using the Django framework. Let's look at some other reasons why you should learn Python Frameworks in Django.

Python Django

Table of Content

Python Django

- Why Use Django Framework?
- Getting Started with Django
- Django Views
- Django URLs
- Django Templates
- Django Models
- Django Forms
- Django Projects
- Online Django Quiz
- Python Django Interview Question

Why Use Django Framework?

- Excellent documentation and high scalability.
- Used by Top MNCs and Companies, such as Instagram, Disqus, Spotify, Youtube, Bitbucket, Dropbox, etc. and the list is never-ending.
- Web framework Django is easy to learn, rapid development, and Batteries fully included.

- The last but not least reason to learn Django in Python, It has a huge library and features such as Web Scraping, Machine Learning, Image Processing, Scientific Computing, etc. One can integrate all this with web applications and do lots and lots of advanced stuff.

Prerequisites to Learn Django

Web framework Django based on Python. You have good knowledge about Python. Some other concepts you should be familiar with are:

Prerequisites to Learn Django

- Understanding of Syntax of Python.
- Understanding of importing and exporting modules is required in the project development phase.
- Understanding Python path concepts to access the data, images or any kind of data.
- Knowledge of Object Oriented concepts as it reduces the code repetition with classes and objects.
- Knowledge about HTML, CSS, JavaScript are very important. As they are the building block of Web development.
- Knowledge about Data Structures like Tuple and List are important.

Getting Started with Django

In this getting started with Django section, you will learn how to get up and running with Django, a powerful web framework for building dynamic websites and applications. Django makes it easy to organize your project into different apps, each handling specific functionality. You'll begin by creating a project, which will automatically set up everything you need, from a folder structure to basic settings.

Getting Started with Django

- Django Introduction and Installation
- When to Use Django? Comparison with other Development Stacks
- Django Project MVT Structure
- Django Basics

- Create an App in Django
- Create a Basic Project using MVT in Django

Getting Started with Django

Once you?ve mastered the basics from this tutorial, theComplete Django Web Development Course ? Basics to Advanceis a perfect next step to dive deeper into both beginner and advanced Django concepts.

Django Views

InDjango viewsare the backbone of handling user requests and rendering responses. There are two primary paradigms for implementing views: Function Based Views (FBVs) and Class Based Views (CBVs). Function Based Views offer simplicity and directness, allowing developers to define views as Python functions. Within this paradigm, common functionalities like creating, listing, displaying details, updating, and deleting objects are implemented as separate functions.

Django Views

While both paradigms have their merits, the choice between FBVs and CBVs ultimately depends on factors such as project requirements, development preferences, and scalability concerns.

Django Views

- Function Based ViewsCreate ViewList ViewDetail ViewUpdate ViewDelete View
- Create View
- List View
- Detail View
- Update View
- Delete View

-	Class	Based	Generic	Views
---	-------	-------	---------	-------

DjangoCreateviewListViewDetailViewUpdateViewDeleteViewFormView

- Createview

- ListView
- DetailView
- UpdateView
- DeleteView
- FormView
- Class Based vs Function Based Views

Django Views

- Create View
- List View
- Detail View
- Update View
- Delete View

Django Views

- Createview
- ListView
- DetailView
- UpdateView
- DeleteView
- FormView

Django URLs

In Django URL patterns serve as a crucial mechanism for directing incoming requests to the appropriate views within your web application. With the flexibility of regular expressions, Django's URL dispatcher allows you to define patterns that match specific URL patterns and route them to corresponding views. When dealing with GET parameters passed through URLs in Django, accessing these parameters within views is straightforward, enabling efficient handling of user inputs

and customization of responses.

Django URLs

- Django URL patterns
- Get parameters passed by urls in Django
- URL Validator in Django
- URL Shortener with Django
- Django URLResolver error

Django Templates

In Django, URLs play a crucial role in navigating through different views and templates within your web application. When working with Django templates, several key concepts enhance the flexibility and functionality of your frontend. Template filters allow you to manipulate variables displayed in your templates, enabling transformations like date formatting or string manipulation.

Django Templates

- Template Filters
- Template Tags
- variables
- Boolean Operators
- for loop
- if ? Django Templates
- Template Inheritance

Django Models

Django Models serve as the backbone of database operations, facilitating seamless management of data. This guide delves into various aspects of Django Models, starting from the fundamental operations of inserting, updating, and deleting data using the Object-Relational Mapping (ORM) provided by Django.

Django Models

Here, You'll learn how to create a basic app model, initialize migrations, and execute them to synchronize your database schema. Moreover, we delve into built-in field validations, ensuring data integrity and consistency, while also delving into the customization of these validations to suit specific application requirements.

Django Models

- ORM ? Inserting, Updating & Deleting Data
- Basic App Model ? Makemigrations and Migrate
- model data types and fields list
- Add the slug field inside Django Model
- Intermediate fields in Django
- Uploading images in Django
- Render Model in Django Admin Interface
- Change Object Display Name using `__str__` function ? Django Models
- Built-in Field Validations ? Django Models
- Custom Field Validations in Django Models
- How to use Django Field Choices ?
- Overriding the save method ? Django

Django Forms

To start, you can create a form using Django Forms by defining a class that inherits from Django's `forms.Form` class. Within this class, you can specify the fields you want to include in your form using various field types provided by Django, such as `CharField`, `IntegerField`, `EmailField`, etc. Once you've defined your form, you can render HTML forms in Django using both GET and POST methods. Django's built-in template tags and filters make it easy to render forms in your HTML templates while ensuring security and CSRF protection.

Django Forms

Django Forms offer a wide range of field types to cater to different data types and validation requirements. Additionally, you can customize the appearance and behavior of form fields by using form field custom widgets, allowing you to enhance user experience and tailor forms to your specific needs.

Django Forms

- How to create a form using Django Forms ?
- Render HTML Forms (GET & POST) in Django
- Django Form Fields
- form field custom widgets
- Initial form data ? Django Forms
- ModelForm ? Create form from Models
- Render Form Fields Manually
- Django Formsets
- Django ModelFormSets

Misc

- Handling Ajax request in Django
- User groups with Custom permissions in Django
- Django Admin Interface
- Extending and customizing django-allauth
- Django ? Dealing with warnings
- Sessions framework using django
- Django Sign Up and login with confirmation Email

Misc

After completing the tutorial and building some projects you might be interested in starting your

career in Django development. We have provided a guide, that will help you in building your career as Django developer.

Django Projects

In this section, we'll explore how to structure and manage Django projects, which are the core framework for building any web application. These projects act as containers for multiple apps that handle specific functionalities, such as user authentication, blog management, or e-commerce operations.

Django Projects

- Google authentication and Fetching mails from scratch
- ToDo webapp using Django
- Django News App
- Weather app using Django
- College Management System Using Django
- E-Commerce Website Using Django
- Creating Word Counter App Using Django
- Youtube Video Downloader Using Django
- Voting System Project Using Django

Online Django Quiz

Test your Django knowledge by answering quiz questions. The quiz questions are meant to test your understanding of Django concepts.

Online Django Quiz

Take Django Quiz

Python Django Interview Question

Interviews are most important aspect of job recruitment and you need to prepare for interviews if you

want to get job sooner. We have compiled some of the most asked interview questions for Django Developers.

Python Django Interview Question

Visit the page [Top 50 Django Interview Questions and Answers](#) to check for interview questions.

Features of Django

- Rapid Development: Django's DRY principle accelerates development by reducing code repetition.
- Admin Interface: Comes with a ready-to-use, customizable admin panel for easy backend management.
- Scalable: Built to handle high traffic and complex applications, ideal for projects of any size.
- Security: Offers built-in protections against common security threats like XSS, CSRF, and SQL injection.
- ORM: Simplifies database interaction using Python, eliminating the need for raw SQL.
- URL Routing: Clean, readable URLs with easy mapping to views.
- Template Engine: Separates logic from presentation for dynamic, reusable web pages.
- Extensive Documentation: Well-organized resources for troubleshooting and learning.
- Active Community: Large community support with abundant third-party plugins and tools.

Applications of Django

Django is a versatile web framework used in a wide range of industries and projects. Here are some common applications:

Applications of Django

- Content Management Systems (CMS): Django is ideal for building custom CMS platforms due to its modularity and flexibility.
- E-commerce Sites: Platforms like e-commerce websites benefit from Django's scalability and robust security features.

- Social Networking Platforms: Django's ability to handle high traffic makes it perfect for social media apps and community-based websites.
- Data-Driven Applications: With its powerful ORM and database management capabilities, Django is great for building applications that rely on large datasets.
- API Development: Django, coupled with Django REST Framework (DRF), makes it simple to develop robust and scalable APIs.
- Scientific Computing Platforms: Django is used in platforms that require complex data analysis and visualization.
- News & Publishing Platforms: Its ability to manage large volumes of content efficiently makes it a go-to for news websites and online publications.
- Educational Platforms: Many e-learning websites and educational tools are built with Django for its scalability and security.

Django vs. Other Web Frameworks

Feature | Django | Java Spring Boot | Express.js

Language | Python | Java | JavaScript (Node.js)

Architecture | Full-stack (MVT) | Full-stack (MVC) | Minimalist

Admin Interface | Built-in admin interface | No built-in admin interface | No built-in admin interface

Development Speed | Fast (due to built-in features) | Medium (more setup required) | Fast (simple routing)

Scalability | Highly scalable | Highly scalable | Highly scalable

Security | Excellent (with built-in protections) | Excellent (strong security features) | Needs additional security measures

ORM | Powerful ORM (built-in) | Powerful ORM (Hibernate) | No built-in ORM

Flexibility | Moderate (convention over configuration) | High (highly configurable) | High (minimalist with custom flexibility)

Documentation | Extensive and detailed | Extensive and detailed | Good but less detailed

Best for | Full-scale apps, CMS, e-commerce | Enterprise-level applications, APIs | Real-time apps, APIs, microservices

Learning Curve | Moderate (due to its many features) | Steeper (requires understanding of Java ecosystem) | Moderate

Use Cases | Social networks, CMS, e-commerce | Large-scale enterprise applications | Real-time apps, microservices

Careers with Django

Here's a table showcasing some common Django career roles along with their approximate salary ranges in both INR and USD:

Careers with Django

Career Role | Salary (INR/year) | Salary (USD/year)

Django Developer | ₹4,00,000 - ₹10,00,000 | \$50,000 - \$90,000

Full-Stack Developer | ₹6,00,000 - ₹15,00,000 | \$60,000 - \$120,000

Software Engineer | ₹5,00,000 - ₹12,00,000 | \$70,000 - \$110,000

DevOps Engineer | ₹8,00,000 - ₹18,00,000 | \$80,000 - \$140,000

Technical Lead/Architect | ₹12,00,000 - ₹25,00,000 | \$100,000 - \$180,000

What is Django?

Django is a high-level Python web framework that allows developers to build robust, scalable, and secure web applications quickly and efficiently. It follows the Model-View-Controller (MVC) architecture pattern, though it is often referred to as Model-View-Template (MVT) in Django terminology.

Is Django easy to learn

Web framework Django is relatively easy to learn, if you have prior experience with Python.

Is Django for frontend or backend?

Django is a open-source framework that is used for backend development of web application.

What are the Features of Django ?

- Versatile which allows us to develop any kind of web page.
- It is scalable
- It is extremely fast.
- Secure thereby helping developers.
- It comes along with content administrations, authentications.

What is Django Architecture ?

Django is based on MVT(Model View Template architecture) which is based on the MVC(Model View Controller architecture). The common difference between them is that Django take care of controller part.

What is the difference between Flask and Django ?

Django | Flask

Supports large projects. | Supports smaller projects.

Templates, Admin and ORM is built-in. | Templates, Admin and ORM requires to be installed.

Not easy to as compare to Flask. | It is easy to learn.

Complete Web development no need any third party tools. | User can choose any third party tools according to their needs.

Does not support visual debugging. | Supports visual debugging.

Inbuilt bootstrapping tool . | Bootstrapping tools are not available.

Name some companies that uses Django ?

Some companies that uses Django are: Instagram, DISCUS, Mozilla Firefox, Youtube, Instagram, Reddit etc. are using web framework Django.

Name some companies that uses Django ?

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our **Mastering Django Framework - Beginner to Advanced Course** on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through hands-on projects and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

Name some companies that uses Django ?

- Django
- Python Django
- Spotlight
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners

- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025

- GATE DA Course 2025

Similar Reads

Introduction

- What is Docker?
- Features of Docker
- Architecture of Docker
- What is Docker Hub?
- What is Docker Cloud?

Docker Installation

- Docker - Installation on Windows
- How to Install Docker using Chocolatey on Windows?
- How to Install and Configure Docker in Ubuntu?
- How to Install Docker on MacOS?
- How to install and configure Docker on Arch-based Linux Distributions(Manjaro) ?
- How to Install Docker-CE in Redhat 8?

Docker Commands

- Docker Commands
- How to Install Docker using Chocolatey on Windows?
- How to Install and Configure Docker in Ubuntu?
- How to Install Docker on MacOS?
- How to install and configure Docker on Arch-based Linux Distributions(Manjaro) ?
- How to Install Docker-CE in Redhat 8?

Docker Commands

- Docker Commands
- Running Commands Inside Docker Container
- Docker - USER Instruction

Docker Images

- What is Docker Image?
- Working with Docker Images
- Docker - Publishing Images to Docker Hub
- Docker Commit
- Docker - Using Image Tags
- Next.js Docker Images
- How to Use Local Docker Images With Minikube?

Docker Compose

- Docker Compose
- Docker Compose Tool To Run a Multi Container Applications

Docker Engine, Storage

- Data Storage in Docker
- Data Storage in Docker
- Backing up a Docker Container
- Using CLI to Manage Docker Volumes

Docker Networking

- Docker Networking
- Docker - Managing Ports
- Creating a Network in Docker and Connecting a Container to That Network
- Connecting Two Docker Containers Over the Same Network
- How to use Docker Default Bridge Networking?
- Create your own secure Home Network using Pi-hole and Docker

Docker Registry

- What is Docker Registry?
- Docker - Using Public Repositories To Host Docker Images

- Docker - Private Registries
- Creating a Private Repository and Push an Image to That Private Repository
- Docker - Using Public Repositories To Host Docker Images

Docker Containers and Managing Containers

- Containerization using Docker
- Virtualisation with Docker Containers
- Docker - Docker Container for Node.js
- Docker - Remove All Containers and Images
- How to Push a Container Image to a Docker Repository?
- Docker - Container Linking
- How to Manage Docker Containers?
- Mounting a Volume Inside Docker Container
- Difference between Docker Image and Container
- Difference between Virtual Machines and Containers
- How to Install Linux Packages Inside a Docker Container?
- Copying Files to and from Docker Containers
- How to Run MongoDB as a Docker Container?
- Docker - Docker Container for Node.js
- Docker - Container for NGINX
- How to Provide the Static IP to a Docker Container?

Docker Swarm

- Docker Swarm Mode
- Docker Swarm vs Kubernetes

Miscellaneous

- Containerizing Spring Boot Application

- How to Create a PHP Docker Container?
- How to Run a Python Script using Docker?
- How to Dockerize an ExpressJS App ?
- How To Dockerize a ReactJS App?
- Kubernetes vs Docker
- Docker - Deploying WebApps on Docker
- Docker - Continuous Integration
- Difference Between Vagrant and Docker
- How to Setup Jenkins in Docker Container?

Docker Tutorial

-
-
-

Docker Tutorial

In this Docker Tutorial, you'll learn all the basic to advanced concepts like Docker installation, Docker container, Docker commands, Docker run, Docker images, Docker compose, Docker engine, Docker networking, etc.

Docker Tutorial

Docker is a powerful tool used for developing, packaging, and deploying applications efficiently. Docker is a container management service. Docker was released in 2013. It is open-source and available for different platforms like Windows, macOS, and Linux. Docker is quickly shipping, testing, and deploying code.

Docker Tutorial

So that it reduces your delay between writing code and running it in production. You can create self-contained environments known as containers. That can run consistently on different platforms.

Docker Tutorial

Docker Tutorial

You need to install the Docker engine on your computer or device. The Docker engine allows you to create and manage docker containers, docker images, docker hub, docker desktop, etc.

Docker Tutorial

Table of Content

Docker Tutorial

- Introduction
- Docker Installation
- Docker Commands
- Docker Run
- Docker Images
- Docker Compose
- Docker Engine, Storage
- Docker Networking
- Docker Registry
- Docker Containers and Managing Containers
- Docker Swarm
- Miscellaneous
- Why to learn Docker?
- Features of Docker
- Advantages of Docker
- Disadvantages of Docker

Introduction

If you are here, then it very obvious that you want to learn the Docker, so here in this section we

have listed all the basic intro of Docker.

Introduction

- Introduction to Docker
- Features of Docker
- Docker Architecture
- Docker Hub
- Docker Cloud

Docker Installation

This section will walk you through the steps of installing Docker on your system, whether it's Windows, macOS, or Linux. We'll cover the different installation methods, system requirements, and post-installation configurations to ensure you have Docker up and running smoothly, ready to build and run your containerized applications.

Docker Installation

- Docker Installation on Windows
- Docker Installation on Windows using Chocolatey
- Docker Installation on Ubuntu
- Docker Installation on macOS
- Docker Installation on Arch-based Linux
- Docker Installation on RedHat

Docker Commands

In this part of the Docker tutorial we will dive into the essential commands you need to navigate the world of Docker. Whether you're building, running, or managing containers, this guide equips you with the tools to control your containerized applications.

Docker Commands

- Docker Instructions
- Run Commands Inside Containers
- USER Instruction

Docker Run

Docker Run refers to the command used in Docker to create and start containers based on Docker images. It's a fundamental aspect of working with Docker, allowing users to deploy applications and services quickly and efficiently within isolated environments known as containers. Here in this section, you will get to know all the details about the Docker Run.

Docker Run

- Dockerfile
- Syntax of Dockerfile
- How to Build a Web Server Docker File?

Docker Images

A Docker Image contains everything a container needs to run, including the application code, libraries, dependencies, and the operating system it needs.

Docker Images

- Docker Images
- Working with Images
- Publish Docker Images in Hub
- Create a Customized Image
- Use Image Tags
- How to use Next.js Image
- How to use Local Image with Minikube

Docker Compose

Managing multiple containers can get complex! Docker Compose simplifies this process. In this section we have listed down all the details about the Docker Compose like intro and Compose tools.

Docker Compose

- Introduction to Docker Compose
- Compose tools to run Multi Container Apps

Docker Engine, Storage

Docker Engine, also known as Docker Daemon, is the core component of the Docker platform responsible for running and managing Docker containers. Explore this section to get all details about Docker Engine.

Docker Engine, Storage

- Docker Storage
- Docker Data Storage
- Backup Docker Container
- Manage Volumes using CLI

Docker Networking

Docker Networking refers to the set of mechanisms and technologies Docker provides for communication between Docker containers, as well as between containers and the outside world. Go through this section to get more details about Docker Networking.

Docker Networking

- Docker Networking
- Docker Ports
- Creating a Network and connecting a Container
- Connecting Two Docker Containers Over the Same Network
- Default Bridge Networking

- Create your own secure Home Network using Pi-hole and Docker

Docker Registry

After knowing the Docker Networking, in Docker tutorial, in this section we are going to discuss Docker Registry which is refer as central repository for storing the and managing Docker image.

Docker Registry

- Docker Registry
- Docker ? Public Repositories
- Docker ? Private Registries
- Creating a Private Repository and Push an Image to That Private Repository
- Using Public Repositories To Host Docker Images

Docker Containers and Managing Containers

Explores this section to get to know about the fundamental concepts and practical aspects of utilizing Docker technology to deploy and manage software applications efficiently in this section.

Docker Containers and Managing Containers

- Containerization using Docker
- Docker Container Virtualization
- Docker Container for Node.js
- Remove Containers
- Push a Container Image to a Docker Repository
- Docker Container Linking
- Manage Containers using CLI
- Mount Volume inside Container
- Difference between Images and Container
- Difference between Virtual Machines and Containers
- How to Install Linux Package in Docker Container

- Copying Files to and from Docker Containers
- Run MongoDB as Container
- Container for Node.js
- Container for NGNIX
- How to Provide the Static IP to a Docker Container?

Docker Swarm

Docker Swarm steps in as your reliable organizer, making sure all your containers are in the right place and working well together. Go through this section to get an all about Docker Swarm.

Docker Swarm

- Introduction to Docker Swarm
- Difference between Kubernetes and Docker Swarm

Miscellaneous

- Running a Java Application using Docker
- Running a PHP Application using Docker
- Running a Python Application using Docker
- How to Dockerize an ExpressJS App?
- How to Dockerize a ReactJS App?
- Docker Vs Kubernetes
- Deploying WebApps on Docker
- Docker Continuous Integration
- Difference Between Vagrant and Docker
- How to Setup Jenkins in Docker Container?

Miscellaneous

Docker Container: A Docker container is a lightweight and executable package of software. It includes everything used to run an application, code, libraries, and dependencies. You don?t need

to allocate any memory for the application. It can automatically generate space according to the requirements.

Why to learn Docker?

A couple of years back, when organizations needed other applications, they buy a server without knowing the performance requirement of the software/application. This results waste of money and resources. Then Virtual machines come onto the market, which allows engineers to run multiple applications on the same resource but as a completely different server. But Every application to run on a Virtual machine requires OS and every OS needs its own CPU, RAM, etc. to run. Which increases the cost.

Why to learn Docker?

Then Docker Model comes into the scope, which overcomes the drawbacks of the Virtual Machine. It reduces the wastage of resources by sharing OS, memory, and CPU, and It offers many benefits for developers and system administrators, like consistency, portability, efficiency, security, scalability, and version control. These benefits make it easier to manage and deploy applications.

Features of Docker

- Docker reduces the size of development by providing a smaller part of the OS via containers.
- It is easier to work on the same project by different teams with the help of Containers.
- Docker containers can be deployed anywhere, on any physical, or virtual machines and on the cloud.
- Docker containers are lightweight so, it becomes easy to scale them.

Advantages of Docker

- Docker uses less memory.
- The full operating system is not required to run an application.
- Containers run faster than the other Virtual Machines.
- It is lightweight.

- It allows us to use a remote repository to share your containers with other teams.
- To reduce the risks, it uses dependencies.

Disadvantages of Docker

- Complexity will increase due to layering.
- It is difficult to manage a large number of containers.
- For an application that needs better graphics, Docker is not suitable for it.
- Cross-platform compatibility is not allowed.

Conclusion

Docker is a software development framework that allows you to package and run apps inside virtual machines on a server. It's often used to create and test applications before deploying them on a real, physical server. This allows organizations to test and deploy their applications quickly and with minimal resources. Docker also makes it easy to package and update apps on any server, regardless of its hardware configuration.

How does Docker work?

Docker uses containerization technology to create and deploy applications in a consistent and isolated environment. It utilizes a client-server architecture where the Docker client interacts with the Docker daemon to build, run, and manage containers.

What are the benefits of using Docker?

Using Docker can provide benefits such as improved developer productivity, increased application portability, efficient resource utilization, and simplified deployment and scaling of applications.

Is Docker the same as virtualization?

No, Docker uses containerization, which is a lightweight form of virtualization. Unlike traditional virtualization, containers share the host OS kernel and do not require a separate guest OS for each application.

How can I get started with Docker?

To get started with Docker, you can install Docker Desktop or Docker Toolbox based on your operating system. Then, you can create, build, and run your first Docker container using the available documentation and tutorials.

Can Docker be used in production environments?

Yes, Docker is widely used in production environments to streamline application deployment, facilitate continuous integration and continuous delivery (CI/CD), and enable efficient management of microservices-based architectures.

Can Docker be used in production environments?

Can Docker be used in production environments?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the Three 90 Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

Can Docker be used in production environments?

A

Can Docker be used in production environments?

- DevOps
- Docker
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++

- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System

- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics

- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates

- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD

- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course

- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

FastAPI ? Introduction

-
-
-

FastAPI ? Introduction

Developers are continuously on the lookout for technologies that allow them to rapidly and efficiently construct sophisticated APIs and online applications. FastAPI, a relatively new addition to the Python web framework landscape, has quickly garnered traction due to its speed, simplicity, and developer-friendly features. In this article, we will see the introduction of FastAPI and explain why it has become a popular alternative for developing web applications and APIs. To learn more about API please refer to [What is API](#).

What is FastAPI?

FastAPI is a modern web framework that is relatively fast and used for building APIs with Python 3.7+ based on standard Python-type hints. FastAPI also assists us in automatically producing documentation for our web service so that other developers can quickly understand how to use it. This documentation simplifies testing web service to understand what data it requires and what it offers. FastAPI has many features like it offers significant speed for development and also reduces human errors in the code. It is easy to learn and is completely production-ready. FastAPI is fully compatible with well-known standards of APIs (i.e. OpenAPI and JSON schema).

Features of FastAPI

- Automatic Documentation: FastAPI generates interactive API documentation automatically using the OpenAPI standard. You can access this documentation by visiting a specific endpoint in your application, which makes it incredibly easy to understand and test your API without having to write extensive documentation manually.
- Python Type Hints: One of FastAPI's standout features is its use of Python-type hints. By

annotating function parameters and return types with type hints, you not only improve code readability but also enable FastAPI to automatically validate incoming data and generate accurate API documentation. This feature makes your code less error-prone and more self-documenting.

- **Data Validation:**FastAPI uses Pydantic models for data validation. You can define your data models using Pydantic's schema and validation capabilities. This ensures incoming data is automatically validated, serialized, and deserialized, reducing the risk of handling invalid data in your application.

- **Asynchronous Support:**With the rise of asynchronous programming in Python, FastAPI fully embraces asynchronous operations. You can use Python's `async` and `await` keywords to write asynchronous endpoints, making it well-suited for handling I/O-bound tasks and improving the overall responsiveness of your application.

- **Dependency Injection:**FastAPI supports dependency injection, allowing you to declare dependencies for your endpoints. This helps in keeping your code modular, testable, and maintainable. You can seamlessly inject dependencies like database connections, authentication, and more into your routes.

- **Security Features:**FastAPI includes various security features out of the box, such as support for OAuth2, JWT (JSON Web Tokens), and automatic validation of request data to prevent common security vulnerabilities like SQL injection and cross-site scripting (XSS) attacks.

Installation and Setup of FastAPI

To get started with FastAPI, you need to install Python, if not then install Python3. Then, you need to install fast API using the following command

Installation and Setup of FastAPI

You also need to install uvicorn

Create a Simple API

Here, we are creating a simple web service that says "Hello" when you visit a specific web address.

With FastAPI, you can do this in just a few lines of code, To run this code, you can save it in a Python file, here we are saving the file as main.py.

Python3

Python3

Python3

Python3

```
from fastapi import FastAPI# Create a FastAPI applicationapp=FastAPI()# Define a route at the root web address ("/")@app.get("/")def read_root():return {"message": "Hello, FastAPI!"}
```

Python3

Now, execute the following command in your terminal:

Python3

Once the application is running, open your web browser and navigate to

Python3

You should see a message displayed in your browser or the response if you are using an API testing tool like curl or Postman.

Advantage of FastAPI

Here are simple advantages of using FastAPI:

Advantage of FastAPI

- Easy to Learn and Use:FastAPI is designed to be straightforward, especially for Python developers. Its simple and intuitive syntax, along with automatic documentation generation, makes it easy to get started and maintain.
- High Performance:FastAPI is built for speed. It's one of the fastest Python web frameworks

available, thanks to its asynchronous support and efficient data handling. This means your web applications can handle a large number of requests without slowing down.

- Automatic Data Validation: With FastAPI, you can use Python type hints to define the data structure you expect for your API requests and responses. FastAPI automatically validates the data, reducing the chances of errors caused by incorrect input.
- Authentication and Authorization: It provides simple ways to handle authentication and authorization, whether using OAuth2, JWT tokens, or custom methods.
- Middleware: We can easily add middleware to your FastAPI application for tasks like logging, authentication, or request/response modification.

Disadvantage of FastAPI

Here are some potential disadvantages of using FastAPI:

Disadvantage of FastAPI

- Learning Curve: While FastAPI is designed to be developer-friendly, it may still have a learning curve for those new to asynchronous programming or web frameworks in general. Developers with no prior experience in Python may also need to learn Python first.
- Community and Documentation: Although FastAPI's community is growing rapidly, it may not have as extensive a support network or documentation as some other frameworks. You may encounter fewer tutorials, guides, and community-contributed packages.

Disadvantage of FastAPI

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the 90 Day Challenge? complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

Disadvantage of FastAPI

P

Disadvantage of FastAPI

- Geeks Premier League
- Python
- FastAPI
- Geeks Premier League 2023

Disadvantage of FastAPI

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++

- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter

- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design

- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases

- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Firestore Tutorial

-
-
-

Firestore Tutorial

Firestore is a comprehensive app development platform provided by Google. It offers a suite of tools and services that help developers build, deploy, and grow their mobile (Android/iOS) and web applications quickly and efficiently. The key features of Firestore include a real-time database, authentication, cloud storage, hosting, and cloud functions, which allow developers to focus on building great user experiences without worrying about the underlying infrastructure.

Firestore Tutorial

This Firestore tutorial is for beginners and students. You will learn all Firestore concepts from basics to advanced, such as setting up Firestore, performing CRUD operations, Firestore Authentication, and Cloud Firestore.

Prerequisites

To learn Firestore, you need basic programming skills in JavaScript, Swift, Kotlin, or Java, along with a basic understanding of NoSQL databases and RESTful APIs.

Latest/Upcoming Features in Firestore 2024

Firestore continues to evolve with several exciting updates and new features for 2024:

Latest/Upcoming Features in Firestore 2024

- Firestore Data Connect: Seamlessly integrates with PostgreSQL databases hosted on Google Cloud SQL, enabling direct database interactions without complex API layers, simplifying data management.
- Firestore Genkit: A pioneering AI framework that simplifies the integration of advanced AI features

into apps, providing tools and libraries for easy development and deployment of sophisticated AI functionalities.

- Firestore Enhancements: Adds vector storage and K-nearest neighbors (KNN) queries, ideal for implementing complex AI-driven data sorting and searching features in generative AI applications.
- Vertex AI for Firebase SDKs: Allows developers to invoke AI models directly from their apps across Kotlin, Swift, Dart, and JavaScript, enhancing development flexibility and access to AI functionalities.

Firebase Tutorial

- Introduction to Firebase
- Environment Setup
- Firebase Realtime Database
- Firebase Authentication
- Firebase Cloud Firestore
- Firebase Cloud Functions
- Firebase Hosting
- Firebase Cloud Storage
- Firebase Analytics
- Firebase Performance Monitoring
- Firebase Remote Config

Introduction to Firebase

If you are new to Firebase, then in this section you will get to know all about Firebase like what is Firebase, why to use it, and also find the difference between other databases.

Introduction to Firebase

- What is Firebase
- Why Use Firebase

- Firebase vs. MongoDB
- Firestore and its advantages

Environment Setup

To learn Firebase it is very important, to setup the environment and in the this Firebase tutorial, Setting up the Firebase environment is an essential first step in leveraging its powerful suite of tools and services for your application development. So, here in this section we have describe how to setup the Firebase environment setup.

Environment Setup

- Getting Started with Firebase for the Web
- Add Firebase to Your JavaScript Project

Firebase Realtime Database

In this section, you will read about the Firebase Realtime Database, a cloud-hosted NoSQL database that allows data to be stored and synchronized in real-time across all connected clients. Ideal for applications requiring live updates, such as chat apps or collaborative tools. So explore and get to how data sync in realtime.

Firebase Realtime Database

- Introduction to Firebase Realtime Database
- Setting up Realtime Database
- Data Organization in Firebase Realtime Database
- Firebase - Write Data
- Firebase - Read Data
- Firebase - Event Types
- Firebase - Queries
- Firebase - Detaching Callbacks
- Querying Data in Realtime Database

- Realtime Synchronization
- Firebase - Arrays
- Firebase - Data

Firebase Authentication

In this section, you will get to know how Firebase Authentication simplifies the process of adding secure user authentication to your app. Here you will get an in-depth knowledge about how to implement these authentication methods, manage user accounts, and handle secure login sessions, ensuring a seamless and secure experience for your app's users.

Firebase Authentication

- What is Firebase Authentication
- Types of Authentications in Firebase
- Email/Password Authentication
- Firebase Email/Password Auth with REST API
- Firebase - Google Authentication
- Firebase - Facebook Authentication
- Firebase - Twitter Authentication
- Firebase - GitHub Authentication
- Anonymous Authentication
- Phone Number Authentication
- Firebase Custom Authentication Using Cloud Functions
- Firebase - Offline Capabilities

Firebase Cloud Firestore

In this section, you will get to know Firebase Cloud Firestore, a flexible and scalable NoSQL database from Firebase. Here you'll learn how to set up Firestore, structure your data, perform complex queries, and leverage its real-time synchronization features to enhance user experiences in

your web and mobile apps.

Firebase Cloud Firestore

- Introduction to Cloud Firestore
- Setting up Cloud Firestore in Your Project
- Data Modeling Basics for Cloud Firestore
- Writing and Reading Data in Cloud Firestore
- How to Create and Add Data to Firebase Firestore in Android
- How to Update Data in Firebase Firestore in Android
- How to Delete Data from Firebase Firestore in Android
- Cloud Firestore ? Add, Set, Update, Delete, Get Data
- How to Perform Queries with Logical OR in Cloud Firestore
- FireSQL - Query Firestore Using SQL Syntax
- Querying and Sorting Firestore Data
- Firestore - Get Realtime Updates
- Structuring Data for Scalability
- Basic Examples of Using Cloud Firestore Security Rules
- What is the Default Security Rule for Firestore?

Firebase Cloud Functions

In this section, you will get to know Firebase Cloud Functions, a serverless framework that lets you run backend code in response to events triggered by Firebase features and HTTPS requests.

Firebase Cloud Functions

- Introduction to Firebase Cloud Functions
- Setting up Firebase Cloud Functions
- Firebase Creating and Deploying Cloud Function
- Firebase Triggers

- Server-Side Code with Firebase Functions
- Integrating with Other Firebase Services
- Integrating with Third-Party Services

Firebase Hosting

In this section, you will learn how to set up Firebase Hosting, deploy your site with a single command using the Firebase CLI, and manage multiple sites and environments effortlessly.

Firebase Hosting

- Introduction to Firebase Hosting
- Setting up Firebase Hosting in Your Project
- Deploying Your Web App
- Custom Domains and SSL
- How to Add a Custom Domain in Firebase Authentication?
- How to Share Firebase Authentication Across Subdomains?
- What is Rollback in Firebase Hosting?

Firebase Cloud Storage

In this section, you will get to know Firebase Cloud Storage, a robust solution for storing and serving user-generated content such as photos, videos, and other large files.

Firebase Cloud Storage

- Introduction to Firebase Cloud Storage
- Getting Started with Cloud Storage on Web
- Download Files from Cloud Storage
- Security Rules for Cloud Storage
- Integrating Cloud Storage with Your App

Firebase Analytics

Firebase Analytics provides detailed insights into user behavior and app performance through comprehensive data collection and reporting. You will learn how to set up and configure Firebase Analytics to track user actions, create custom events, and analyze user engagement.

Firebase Analytics

- Setting up Firebase Analytics in Your Project
- Tracking User Engagement
- Adding Firebase Custom Events for Analytics
- Conversion Tracking with Firebase
- Create and Segment Audiences on Google Firebase

Firebase Performance Monitoring

In this section, you will get to know how Firebase Performance Monitoring helps you track your app's performance.

Firebase Performance Monitoring

- Introduction to Firebase Performance Monitoring
- Tracing App Crashes
- How to Trace Firebase Crashes?

Firebase Remote Config

In this section, you will get to know Firebase Remote Config, a powerful feature that allows you to dynamically change the behavior and appearance of your app without requiring users to download an update.

Firebase Remote Config

- Introduction to Firebase Remote Config
- Setting up Remote Config in Your Project
- Remote Parameter Configuration

- Introduction to Firebase A/B Testing
- A/B Testing with Remote Config
- Configure Conditional Delivery

Advantages of Firebase

Here are some of the advantages of Firebase:

Advantages of Firebase

- Real-Time Data Synchronization: Firebase's real-time database synchronizes data instantly across all clients, making it ideal for chat applications, collaborative tools, and live data feeds.
- Ease of Use: Firebase offers a user-friendly interface and seamless integration, which helps developers get started quickly without needing extensive backend knowledge.
- Comprehensive Suite of Tools: It provides a wide range of services including authentication, cloud storage, hosting, and analytics, all integrated within a single platform.
- Scalability: Firebase automatically scales with the application's needs, handling increases in data and user load without requiring additional configuration.
- Serverless Architecture: With Cloud Functions, Firebase allows developers to run backend code in response to events triggered by Firebase features and HTTPS requests, without managing servers.
- Cross-Platform Support: Firebase supports web, iOS, and Android platforms, enabling a consistent development experience across different devices.

Firestore vs. Firebase

Aspects | Firestore Realtime Database | Firestore

Database Structure | JSON tree structure | Collections and documents

Querying | Limited to sorting and filtering | Supports compound queries, filtering, sorting, pagination

Scalability | Suitable for smaller applications | Designed for scalability and larger applications

Real-time Updates | Yes | Yes

Data Model | JSON-based | NoSQL, collections, and documents

Firebase vs. Firestore

For more: [Difference between Firebase and Firestore](#)

Conclusion

Firebase emerges as a dynamic platform, offering a comprehensive suite of tools that streamline app development. Through this Firebase tutorial, you will explore Firebase versatile features, from real-time database management to seamless user authentication. Firebase empowers developers to focus on crafting exceptional user experiences by eliminating the complexities of backend infrastructure.

What is Firebase?

Firebase is a software development platform offered by Google, providing a suite of tools and services to streamline app development. It encompasses features such as real-time database management, user authentication, cloud messaging, hosting, and more.

How can I get started with Firebase?

To get started with Firebase, visit the Firebase website and sign in with your Google account. From there, you can create a new Firebase project and access a range of services to integrate into your app.

Is Firebase suitable for both Android and iOS app development?

Yes, Firebase supports both Android and iOS app development, providing SDKs and tools for seamless integration into apps on both platforms.

Can I use Firebase for user authentication in my app?

Yes, Firebase Authentication offers various methods for user authentication, including email/password authentication, social authentication using providers like Google, Facebook, and Twitter, as well as phone number authentication.

Is Firebase suitable for real-time collaborative applications?

Yes, Firebase's real-time database and cloud messaging capabilities make it ideal for building real-time collaborative applications such as chat apps, multiplayer games, and collaborative document editing tools

Is Firebase suitable for real-time collaborative applications?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Is Firebase suitable for real-time collaborative applications?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Is Firebase suitable for real-time collaborative applications?

V

Is Firebase suitable for real-time collaborative applications?

- Firebase
- Databases

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution

- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures

- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs

- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker

- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy

- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced

- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course

- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE

- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Flask Setup & Installation

- Introduction to Web development using Flask
- Differences Between Django vs Flask
- How to Install Flask in Windows?

Flask Quick Start

- Flask - (Creating first simple application)
- How to Run a Flask Application
- Flask App Routing
- Flask - HTTP Method
- Flask - Variable Rule
- Redirecting to URL in Flask
- Python Flask - Redirect and Errors
- How to Change Port in Flask app
- Changing Host IP Address in Flask

Serve Templates and Static Files in Flask

- Flask Rendering Templates
- CSRF Protection in Flask
- Template Inheritance in Flask
- Placeholders in jinja2 Template - Python
- How to serve static files in Flask
- Uploading and Downloading Files in Flask
- How to Upload File in Python-Flask
- Upload Multiple files with Flask
- Flask - Message Flashing
- Create Contact Us using WTForms in Flask
- Sending Emails Using API in Flask-Mail

User Registration, Login, and Logout in Flask

- How To Add Authentication to Your App with Flask-Login
- Add User and Display Current Username in Flask
- Password Hashing with Bcrypt in Flask
- How to store username and password in Flask
- Flask - Role Based Access Control
- How to use Flask-Session in Python Flask ?
- Flask Cookies
- How to return a JSON response from a Flask API ?

Define and Access the Database in Flask

- Flask and SQLAlchemy Tutorial for Database
- How to Build a Web App using Flask and SQLite in Python
- Sending Data from a Flask app to MongoDB Database
- Making a Flask app using a PostgreSQL database
- Login and Registration Project Using Flask and MySQL
- How to execute raw SQL in Flask-SQLAlchemy app

Flask Deployment and Error Handling

- Subdomain in Flask | Python
- Handling 404 Error in Flask
- Deploy Python Flask App on Heroku
- Deploy Machine Learning Model using Flask

Flask Tutorial

-
-
-

Flask Tutorial

This Flask tutorial is the latest and comprehensive guide designed for beginners and professionals to learn Python Web Framework Flask, one of the most popular Python-based web frameworks. Whether you are a beginner or an experienced developer, this tutorial is specially designed to help you learn and master Flask and build your real-world web applications.

Flask Tutorial

This Flask tutorial covers all the details of Flask, from basic concepts such as setup and installation to advanced concepts like user authentication, database integration, and deployment. In addition to this, we also provide you with a list of Flask with Python projects, FAQs, and interview questions for your future Interview.

Flask Tutorial

Flask Tutorial

Flask Tutorial

Before you jump into this tutorial, it's recommended to have a hands-on experience with HTML and Python. If you are not well aware of these concepts, then we will suggest you go through our tutorials on HTML and Python.

Flask Tutorial

Table of Content

Flask Tutorial

- What is Flask?
- Flask Setup & Installation
- Flask Quick Start
- Serve Templates and Static Files in Flask
- User Registration, Login, and Logout in Flask

- Define and Access the Database in Flask
- Flask Deployment and Error Handling
- Flask Projects

Prerequisites for the Flask Tutorial

Before getting deep into this tutorial, we assume that you already have some experience on HTML and Python. And if you are not familiar with these concepts, we recommend checking out our brief tutorials on [HTML](#) and [Python](#) first.

Flask Setup & Installation

Welcome to the Flask tutorials section! Here, we'll cover the essential elements you need to kickstart your journey in Flask. From installation and keywords to comments, variables, and indentation, we'll explore the foundational concepts that underpin Python with Flask development.

Flask Setup & Installation

- Introduction to Web Development using Flask
- Differences Between Django vs Flask
- Installation of Flask on Windows

Flask Quick Start

Welcome to Flask Quick Start ? your go-to guide for mastering Flask development essentials in a snap! From crafting your inaugural application to fine-tuning HTTP methods, routes, and redirects, this concise yet comprehensive resource equips you with the skills needed to excel in Flask development.

Flask Quick Start

- Creating the first simple application
- Run a Flask Application
- Flask Routes

- Flask Models
- Flask ? HTTP Method
- Flask ? Variable Rules
- Redirects and URL
- Redirect & Errors
- Change Port in Flask app
- Changing Host IP Address in Flask

Serve Templates and Static Files in Flask

Discover the essential techniques for serving templates and static files in Flask Web Framework with this comprehensive guide. From mastering Jinja2 templating and template inheritance to ensuring CSRF protection, uploading and managing files, and leveraging Flask Web Framework extensions like Flask-Mail and Flask WTF, this resource covers all you need to know to enhance your Flask web Framework applications.

Serve Templates and Static Files in Flask

- Templates in Flask
- CSRF Protection in Flask
- Templating With Jinja2 in Flask
- Template Inheritance in Flask
- Placeholders in Jinja2 Template
- Flask ? Static Files
- Uploading and Downloading Files in Flask
- Upload File in Python-Flask
- Upload Multiple files with Flask
- Flask ? Message Flashing

- Flask WTF
- Flask-Mail

Serve Templates and Static Files in Flask

- CSRF Protection in Flask
- Templating With Jinja2 in Flask
- Template Inheritance in Flask
- Placeholders in Jinja2 Template

Serve Templates and Static Files in Flask

- Upload File in Python-Flask
- Upload Multiple files with Flask

User Registration, Login, and Logout in Flask

Master user authentication in Flask Web Framework with this concise guide. Learn registration, login, and logout functionalities using Flask-Login, implement password hashing with Bcrypt, manage sessions, and explore role-based access control. Additionally, discover tips for displaying the current username and handling JSON responses effortlessly.

User Registration, Login, and Logout in Flask

- How To Add Authentication to Your App with Flask-Login
- Display Current Username in Flask
- Password Hashing with Bcrypt in Flask
- How to store username and password in Flask
- Flask ? Role-Based Access Control
- Flask Sessions to server Logout
- Flask ? JWT
- Flask Cookies
- Flask ? JSON Response

Define and Access the Database in Flask

Delve into the world of database management in Flask with Python this guide. Explore powerful tools like Flask-SQLAlchemy for seamless integration with relational databases like SQLite and PostgreSQL. Learn to send data to MongoDB and execute raw SQL commands with Flask-SQLAlchemy.

Define and Access the Database in Flask

- Flask-SQLAlchemy
- Flask SQLite
- Sending Data from a Flask app to MongoDB Database
- Making a Flask app using a PostgreSQL Database
- Build a Web App using Flask and SQLite in Python
- Login and Registration Project Using Flask and MySQL
- Execute raw SQL in the Flask-SQLAlchemy

Flask Deployment and Error Handling

Delve into Flask with Python deployment and error handling, covering subdomain management, 404 error handling, deploying Flask apps on Heroku, and incorporating machine learning models seamlessly.

Flask Deployment and Error Handling

- Subdomain in Flask
- Handling 404 Error in Flask
- Deploy Python Flask App on Heroku
- Deploy Machine Learning Model using Flask

Flask Projects

Explore a variety of Flask Web Framework projects, from a todo list app to sentiment analysis on Twitter, showcasing diverse functionalities such as portfolio creation, MySQL integration, and

interactive data visualization with graphs and charts.

Flask Projects

- Todo list app using Flask Python
- Single Page Portfolio Using Flask
- Profile Application using Python Flask and MySQL
- Wikipedia search app using Flask Framework ? Python
- Twitter Sentiment Analysis WebApp Using Flask
- Create Cricket Score API using Web Scraping in Flask
- How to Add Graphs to Flask AppsAdd Scatter Chart in FlaskShow Data on Google Pie Chart using Python Flask
- Add Scatter Chart in Flask
- Show Data on Google Pie Chart using Python Flask
- Flask Projects Archives

Flask Projects

- Add Scatter Chart in Flask
- Show Data on Google Pie Chart using Python Flask

Difference Between Flask and Django

Here in this section, we have create a comparison table of Flask and Django.

Difference Between Flask and Django

Feature | Flask | Django

Framework Type | Micro-framework (lightweight, minimal setup) | Full-stack framework (includes built-in features)

Flexibility | Highly flexible, allows customization | Less flexible but provides a lot of built-in tools

Development Speed | Slower for larger projects due to less automation | Faster for large projects with built-in solutions

Learning Curve | Easier for beginners due to simplicity | Steeper learning curve due to more complex features

Use Case | Best for small to medium web applications | Best for large, complex applications

Project Structure | Simple, user-defined structure | Follows a specific, organized project structure

Built-in Features | Minimal, requires third-party libraries | Comes with built-in features like ORM, admin panel, and authentication

Scalability | Requires more manual work for scalability | High scalability with built-in tools for large projects

ORM Support | Optional (Flask-SQLAlchemy for database support) | Built-in ORM (Django ORM)

Template Engine | Jinja2 | Django Template Language

Security Features | Requires manual implementation of many security features | Comes with built-in security features like CSRF protection, user authentication

URL Routing | Manual, more customizable | Automatic URL routing with pre-defined patterns

Career Opportunities with Flask

Career Role | Salary (INR) | Salary (USD)(Approx.)

Entry-Level Flask Developer | ₹3,00,000 ? ₹5,00,000 per annum | \$3,600 ? \$6,000 per annum

Mid-Level Flask Developer | ₹5,00,000 ? ₹9,00,000 per annum | \$6,000 ? \$10,800 per annum

Senior Flask Developer | ₹9,00,000 ? ₹15,00,000 per annum | \$10,800 ? \$18,000 per annum

Lead Developer/Architect | ₹15,00,000 ? ₹25,00,000 per annum | \$18,000 ? \$30,000 per annum

Career Opportunities with Flask

Must Read ? Top 40 Flask Interview Questions and Answers

What is WSGI?

WSGI is an acronym for web server gateway interface which is a standard for Python web framework flask application development. It is considered the specification for the universal interface between the web server and web application.

What is Jinja2?

Jinja2 is a web template engine that combines a template with a certain data source to render dynamic web pages. In Easy Language, It combines a template (the layout of the page) with data (the specific information you want to show) to create a dynamic web page.

Conclusion

This Flask tutorial is a complete guide designed to help both beginners and experienced developers learn and master Flask. It covers everything you need, from getting started with installation to building real-world web applications. You've also learned about user authentication, database connections, and even how to deploy your Flask projects. With practical projects, interview tips, and FAQs included, this tutorial provides everything you need to confidently use Flask to create dynamic web applications. Now you're ready to start building with Flask and take your web development skills

What is Flask?

Flask is a web framework that allows developers to build lightweight web applications quickly and easily with Flask Libraries. It was developed by Armin Ronacher, leader of the International Group of Python Enthusiasts(POCCO). It is basically based on the WSGI toolkit and Jinja2 templating engine.

Should I learn HTML for Flask?

Both are highly recommended in case you are learning to develop web applications.

Is Flask open source?

Yes, Flask and Django both are Free Open Source, Python-based web frameworks that are used for building web applications.

How do I start a Flask project?

Here's a simplified version of the steps to start a Flask project:

How do I start a Flask project?

- Install Python.
- Install Flask by running `?pip install flask?` in the terminal or command prompt.
- Create a virtual environment by running `?python -m venv myenv?`.
- Activate the virtual environment by running `?myenv\Scripts\activate.bat?` on Windows or `?source myenv/bin/activate?` on macOS or Linux.
- Create a Flask app by importing Flask and creating a new instance of the Flask class.
- Define routes by decorating a function with `?@app.route(?/myurl?)?` and defining the function to return data or HTML.
- Run the app by running `?flask run?` in the terminal or command prompt.

What is the default port of Flask?

By default, many Flask applications run on port 5000.

Which company uses Flask?

There are many major companies employing Flask and that list includes Netflix, Reddit, Airbnb, Lyft, Mozilla, MIT, Uber, Red Hat, Rackspace, Mailgun, Patreon, Samsung, NGINX, 2market, B2W, and Sieve.

What is the salary of a Flask Developer in India?

As per various resources, the average Salary of a Flask Developer was ranging between 9,00,000 INR ? 11,00,000 INR depending upon the experience and skills.

What is the salary of a Flask Developer in India?

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the Three 90 Challenge?complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

What is the salary of a Flask Developer in India?

- Python

- Python Flask
- Tutorials

What is the salary of a Flask Developer in India?

- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos

- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- [DSA](#)
- [Data Structures](#)
- [Algorithms](#)
- [DSA for Beginners](#)
- [Basic DSA Problems](#)
- [DSA Roadmap](#)
- [DSA Interview Questions](#)
- [Competitive Programming](#)

Similar Reads

- [Data Science & ML](#)
- [Data Science With Python](#)
- [Data Science For Beginner](#)
- [Machine Learning](#)

- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD

- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL

- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management

- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course

- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT

- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is GraphQL?

-
-
-

What is GraphQL?

Client and server communication follows a traditional approach of REST APIs which is good but nowadays applications have become more powerful and complex which requires a more flexible approach rather than REST, so here GraphQL comes into the picture. In this article, we will go through all the concepts of GraphQL.

What is GraphQL?

GraphQL is an open-source data query language for APIs and it is a server-side runtime for executing the query. The server's GraphQL runtime takes care of executing the query and ensuring that the right data is fetched and sent back.

What is GraphQL?

It is an alternative to REST, where clients make multiple requests to different endpoints to get the data they require but in GraphQL clients can request exactly the data they need in a single query.

What is GraphQL?

It was developed by Facebook and made open source for the whole world.

What is GraphQL?

Example: Let's take an example, Suppose We have a REST API for a blog. If we want to get a blog post and its author information then We have to make two separate requests to the server:

What is GraphQL?

- one for the blog post.
- another for the author's details.

What is GraphQL?

But In GraphQL, we will request both in one query and It will reduce network overhead.

Key Features of GraphQL

GraphQL has several features that set it apart from traditional REST APIs, offering developers a more flexible and efficient way to manage data. Let's explore these features as follows:

Key Features of GraphQL

- Flexible Queries: Clients can request exactly the data they need, avoiding over-fetching and under-fetching.
- Strongly Typed: GraphQL schemas provide clear data structures and types, reducing runtime errors.
- Real-time Updates: GraphQL supports subscriptions for real-time data interactions.
- Single Endpoint: Unlike REST, GraphQL typically uses a single endpoint for all data requests.
- Introspection: Clients can explore the schema's capabilities through introspection queries.
- Batching: Multiple queries can be sent in a single request to minimize network overhead.
- Efficient for Mobile: GraphQL can be more efficient for mobile devices by reducing data transfer.
- Versioning: It eliminates the need for versioning in APIs, as changes can be made without breaking existing clients.

Key Components of GraphQL

For Better Understanding of GraphQL, let's see some key components of GraphQL which are as follows:

1. Schema

It defines the data types that can be queried and their relationships. GraphQL uses its own language that is Schema Definition Language (SDL) for writing the schema. It is human readable language and It does not depend upon any specific language or framework. Schemas have two main types:

1. Schema

- Queries (for retrieving data)
- Mutations (for modifying data).

2. Types

GraphQL defines custom types to define the structure of data. There are two main types of Type:

2. Types

- Scalar Types: It represents values like integers, strings, booleans, and floats.
- Object Types: It represents complex objects with fields. Fields can be scalars or other object types. For example, A "User" object type with fields like "id", "name", and "email".

3. Queries

It is used to retrieve data from a GraphQL server. It specifies what type of data we want to retrieve from fields of which types. It is similar to GET requests in REST APIs but allows to request exactly the data we need. It is reducing over-fetching or under-fetching.

4. Mutations

It is used to modify data on the server. It can be used for creating, updating, or deleting data. Mutations are similar to POST, PUT, or DELETE requests in REST APIs.

GraphQL basic Schema Design

Let's take an example to understand the basics of GraphQL Schema design

GraphQL basic Schema Design

Example:

Explanation:

We have defined three main types:

Explanation:

- Book
- Query
- Mutation

Explanation:

Book Type:

Explanation:

The Book type represents a book object with three fields:

Explanation:

- id:An ID field, which is non-nullable (! indicates that it cannot be null).
- title:A String field, which is non-nullable.
- author:A String field, which is non-nullable.

Explanation:

Query Type:

Explanation:

It defines two query fields:

Explanation:

- books>Returns a list of Book objects, ! indicates a non-null list of non-null Book objects.
- book(id: ID!):Takes an id argument and returns a single Book object.

Explanation:

Mutation Type:

Explanation:

It defines three mutation fields:

Explanation:

- createBook:Creates a new book with title and author arguments and returns a Book.
- updateBook:Updates an existing book with id, title, and author arguments and returns a Book.
- deleteBook:Deletes a book with the given id and returns a Boolean indicating success.

Conclusion

GraphQL is a powerful and flexible data query language for APIs. It is offering a more efficient alternative to traditional REST APIs. It allows clients to request exactly the data they need and reducing network overhead. It provides features like strong typing, real-time updates, and a single endpoint.

Conclusion

Get IBM Certification and a 90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

Conclusion

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need?Start the challenge right away!

Conclusion

K

Conclusion

- GraphQL
- Geeks Premier League
- Databases
- GraphQL
- Geeks Premier League 2023

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++

- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System

- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics

- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates

- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD

- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course

- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Java Overview

- Introduction to Java
- The Complete History of Java Programming Language
- How to Install Java on Windows, Linux and macOS?
- Setting up Environment Variables For Java
- How JVM Works - JVM Architecture
- JDK in Java
- Differences between JDK, JRE and JVM

Java Basics

- Java Syntax
- Java Hello World Program
- Java Identifiers
- Java Keywords
- Java Data Types
- Java Variables
- Scope of Variables in Java
- Java Operators
- Java User Input

Java Flow Control

- Java if statement
- Java if-else Statement
- Java if-else-if ladder with Examples
- Java For Loop
- For-each loop in Java
- Java while Loop
- Java Do While Loop

- Java Break Statement
- Java Continue Statement
- Java return Keyword

Java Methods

- Java Methods
- How to Call a Method in Java?
- Static Method vs Instance Method in Java
- Access Modifiers in Java
- Command Line Arguments in Java
- Variable Arguments (Varargs) in Java

Java Arrays

- Arrays in Java
- How to Initialize an Array in Java?
- Java Multi-Dimensional Arrays
- Jagged Array in Java
- Arrays class in Java
- Final Arrays in Java

Java Strings

- Java Strings
- Why Java Strings are Immutable?
- Java String concat() Method with Examples
- String class in Java
- StringBuffer class in Java
- Java StringBuilder Class
- String vs StringBuilder vs StringBuffer in Java

Java OOPs Concepts

- Java OOP(Object Oriented Programming) Concepts
- Classes and Objects in Java
- Java Constructors
- Object Class in Java
- Abstraction in Java
- Encapsulation in Java
- Inheritance in Java
- Polymorphism in Java
- Method Overloading in Java
- Overriding in Java
- Java Packages

Java Interfaces

- Java Interface
- Interfaces and Inheritance in Java
- Java Class vs Interfaces
- Java Functional Interfaces
- Nested Interface in Java
- Marker Interface in Java
- Java Comparator Interface

Java Collections

- Collections in Java
- Collections Class in Java
- Collection Interface in Java
- Java List Interface
- ArrayList in Java

- Vector Class in Java
- LinkedList in Java
- Stack Class in Java
- Set in Java
- Java HashSet
- TreeSet in Java
- Java LinkedHashSet
- Queue Interface In Java
- PriorityQueue in Java
- Deque Interface in Java
- Map Interface in Java
- HashMap in Java
- Java LinkedHashMap
- Hashtable in Java
- Java Dictionary Class
- SortedSet Interface in Java with Examples
- Java Comparator Interface
- Java Comparable Interface
- Java Comparable vs Comparator
- Java Iterator

Java Exception Handling

- Java Exception Handling
- Java Checked vs Unchecked Exceptions
- Java Try Catch Block
- Java final, finally and finalize
- throw and throws in Java

- User-Defined Custom Exception in Java
- Chained Exceptions in Java
- Null Pointer Exception in Java
- Exception Handling with Method Overriding in Java

Java Multithreading

- Java Multithreading Tutorial
- Java Threads
- Java Thread Class
- Java Runnable Interface
- Lifecycle and States of a Thread in Java
- Main thread in Java
- Java Thread Priority in Multithreading
- Java Naming a Thread and Fetching Name of Current Thread
- Java Thread.start() vs Thread.run() Method
- Java Thread.sleep() Method
- Java Daemon Thread
- Java Thread Safety and How to Achieve it?
- Thread Pools in Java

Java File Handling

- File Handling in Java
- Java File Class
- Java Program to Create a New File
- Java Program to Write into a File
- Delete a File Using Java
- Java FileReader Class
- Java FileWriter Class

- Java FilePermission Class
- Java FileDescriptor Class

Java Streams and Lambda Expressions

- Java Lambda Expressions
- Java Method References
- Java 8 Stream Tutorial
- Java 8 Features - Complete Tutorial

Java IO

- Java IO : Input-output in Java with Examples
- Java Reader Class
- Java Writer Class
- Java FileInputStream Class
- FileOutputStream in Java
- Ways to Read Input from Console in Java
- Java BufferedOutputStream Class
- Java BufferedReader vs Scanner Class
- Fast I/O in Java in Competitive Programming

Java Synchronization

- Synchronization in Java
- Importance of Thread Synchronization in Java
- Java Method and Block Synchronization
- Java Atomic vs Volatile vs Synchronized
- Java Lock Framework vs Thread Synchronization
- Deadlock in Java Multithreading
- Deadlock Prevention And Avoidance

- Difference Between Lock and Monitor in Java Concurrency
- ReentrantLock in Java

Java Regex

- Regular Expressions in Java
- Java Pattern pattern() Method
- Java Matcher pattern() Method
- java.lang.Character Class Methods | Set 1
- Java Quantifiers

Java Networking

- Java Networking
- TCP/IP Model
- User Datagram Protocol (UDP)
- Difference Between IPv4 and IPv6
- Difference Between Connection-oriented and Connection-less Services
- Socket Programming in Java
- Java ServerSocket Class
- Java URL Class

JDBC

- JDBC (Java Database Connectivity)
- JDBC Drivers
- Establishing JDBC Connection in Java
- Types of Statements in JDBC

Java Memory Allocation

- Java Memory Management
- How are Java Objects Stored in Memory?

- Stack vs Heap Memory Allocation
- Java Virtual Machine (JVM) Stack Area
- How Many Types of Memory Areas are Allocated by JVM?
- Garbage Collection in Java
- JVM Garbage Collectors
- Stack vs Heap Memory Allocation
- Memory leaks in Java

Java Interview Questions

- Java Interview Questions and Answers
- Java Multiple Choice Questions

Java Practice Problems

- Java Programs - Java Programming Examples
- Java Exercises - Basic to Advanced Java Practice Programs with Solutions
- Java Quiz | Level Up Your Java Skills

Java Projects

- Top 50 Java Project Ideas For Beginners and Advanced [Update 2025]
- Number guessing game in Java
- Mini Banking Application in Java
- Java program to convert Currency using AWT
- Tic-Tac-Toe Game in Java
- Design Snake Game
- Memory Game in Java
- How to Implement a Simple Chat Application Using Sockets in Java?
- Image Processing in Java - Face Detection
- Design Media Sharing Social Networking System

- Java Swing | Create a simple text editor

Java Tutorial

-
-
-

Java Tutorial

Java is one of the most popular and widely used programming languages.

Java Tutorial

- Used to develop mobile apps, desktop apps, web apps, web servers, games, and enterprise-level systems.
- Java was invented by James Gosling and Oracle currently owns it. JDK 23 is the latest version of Java.
- Java's syntax is similar to C/C++.
- Popular platforms like LinkedIn, Amazon, and Netflix rely on Java for their back-end architecture, showcasing its stability and scalability across different environments.
- Popularity is so high that 3 Billion+ devices use Java across the world.

Java Basics

- What is Java?
- Download and Install Java?
- JDK vs JRE vs JVM
- Identifiers
- Keywords
- Data Types
- Variables
- Operators

- Decision Making (if, if-else, switch, break, continue, jump)
- Loops
- Quizzes:Java Data Types

Java Methods

- Introduction to Methods
- How to Call Methods?
- Static Methods vs Instance Methods
- Access Modifiers
- Command Line Arguments
- Variable Arguments (Varargs)

Java Arrays

- Introduction to Arrays
- Declare and Initialize Arrays
- Multi-Dimensional Arrays
- Jagged Arrays
- Arrays Class
- Final Arrays
- Java Array Programs
- Quizzes:Java Arrays

Java Strings

- Introduction of Strings
- Why Strings are Immutable?
- Java String Concatenation
- String Class
- StringBuffer Class

- StringBuilder Class
- Strings vs StringBuffer vs StringBuilder
- Java String Programs

Java OOPs Concepts

- What are OOPs Concepts?
- Classes and Objects
- Constructors
- Object Class
- Abstraction
- Encapsulation
- Inheritance
- PolymorphismCompile-Time Polymorphism (Method Overloading)Runtime Polymorphism (Method Overriding)
- Compile-Time Polymorphism (Method Overloading)
- Runtime Polymorphism (Method Overriding)
- Packages
- Quizzes:Constructors,Packages,Classes and Objects

Java OOPs Concepts

- Compile-Time Polymorphism (Method Overloading)
- Runtime Polymorphism (Method Overriding)

Java Interfaces

- Java Interfaces
- Interfaces and Inheritance
- Class vs Interface
- Functional Interface

- Nested Interface
- Marker Interface
- Comparator Interface
- Quizzes:Interface and Abstract Class

Java Collections

- Java Collection Framework
- Collections Class
- Collection Interface
- List InterfaceArrayList ClassVector ClassLinkedList ClassStack Class
- ArrayList Class
- Vector Class
- LinkedList Class
- Stack Class
- Set InterfaceHashSet ClassTreeSet ClassLinkedHashSet Class
- HashSet Class
- TreeSet Class
- LinkedHashSet Class
- Queue InterfacePriority Queue ClassDeque Interface
- Priority Queue Class
- Deque Interface
- Map InterfaceHashMap ClassLinkedHashMap ClassHashTable ClassDictionary
- HashMap Class
- LinkedHashMap Class
- Hashtable Class
- Dictionary
- SortedSet Interface

- Comparator Interface
- Comparable Interface
- Comparator vs Comparable
- Iterator

Java Collections

- ArrayList Class
- Vector Class
- LinkedList Class
- Stack Class

Java Collections

- HashSet Class
- TreeSet Class
- LinkedHashSet Class

Java Collections

- Priority Queue Class
- Deque Interface

Java Collections

- HashMap Class
- LinkedHashMap Class
- Hashtable Class
- Dictionary

Java Exception Handling

- Java Exceptions
- Checked vs Unchecked Exceptions

- Try Catch Block
- Final, Finally and Finalize
- Throw and Throws
- Customized Exception Handling
- Chained Exceptions
- Null Pointer Exceptions
- Exception Handling with Method Overriding
- Quizzes:Exception Handling

Java Multithreading

- Introduction to Java Multithreading
- Threads
- Thread Class
- Runnable Interface
- Lifecycle and States of a Thread
- Main Thread
- Thread Priority in Multithreading
- Naming & Fetching Name of Current Thread
- Thread.start() vs Thread.run() Method
- Thread.sleep() Method
- Daemon Thread
- Thread Safety
- Thread Pools

Java File Handling

- File Handling
- File Class
- Create Files

- Read Files
- Write on Files
- Delete File
- FileReader Class
- FileWriter Class
- FilePermission Class
- FileDescriptor Class

Java Streams and Lambda Expressions

- Lambda Expressions
- Method References
- Java Stream - Complete Tutorial
- Java 8 Features - Complete Tutorial

Java IO

- Introduction to Java IO
- Reader Class
- Writer Class
- FileInputStream
- FileOutputStream
- BufferedReader Input Stream
- BufferedReader Output stream
- BufferedReader vs Scanner
- Fast I/O

Java Synchronization

- Java Synchronization
- Importance of Thread Synchronization

- Method and Block Synchronization
- Atomic vs Volatile vs Synchronized
- Local Frameworks vs Thread Synchronization
- Deadlock in Multithreading
- Deadlock Prevention and Avoidance
- Lock vs Monitor in Concurrency
- Reentrant Lock

Java Regex

- What is Java Regex?
- Pattern Class
- Matcher Class
- Character Class
- Quantifiers

Java Networking

- Introduction to Java Networking
- TCP Architecture
- UDP Architecture
- IPV4 vs IPV6
- Connection-Oriented vs Connectionless Protocols
- Socket Programming
- Server Socket Class
- URL Class and Methods

Java Database Connectivity(JDBC)

- Introduction to Java JDBC
- JDBC Driver

- JDBC Connection
- Types of Statements in JDBC

Java Memory Allocation

- Java Memory Management
- How Java Objects Stored in Memory?
- Stack vs Heap Memory Allocation
- Java Virtual Machine(JVM) Stack Area
- Types of Memory Areas Allocated by JVM
- Garbage Collection
- Types of JVM Garbage Collectors
- Heap and Stack Memory Allocation
- Memory Leaks

Java Interview Questions

- 200+ Core Java Interview Questions and Answers
- Java MCQ

Java Practice Problems

- Java Programming Examples
- Java Exercises ? Basic to Advanced Java Practice Programs
- Java Language MCQs with Answers
- Java Practice Quiz

Java Projects

- Number Guessing Game
- Simple Banking Application
- Currency Converter
- Tic-Tac-Toe Game

- Snake Game
- Memory Game
- Chat Application
- Face Detection System
- Social Networking Site
- Text Editor

Java Jobs & Opportunities

According to report of Statista, Java is most demanded programming languages after JavaScript by recruiters worldwide in 2024. Major MNC companies are recruiting Java Programmers.

Java Jobs & Opportunities

- Amazon
- IBM
- Google
- Honeywell
- Netflix
- Meta
- Wipro
- Infosys
- Accenture
- Oracle Inc
- Deloitte India
- Cognizant
- Capgemini and Many more...

Java Jobs & Opportunities

What is Java?

Developed by Sun Microsystems in 1995, Java is a highly popular, object-oriented programming language. This platform independent programming language is utilized for Android development, web development, artificial intelligence, cloud applications, and much more.

Why use Java ?

Java is simple to understand programming language because doesn't contain concepts like :
Pointers and operator overloading and it is secure and portable.

What are the major concepts in Java?

There are 4 major concept in Java that is abstraction, encapsulation, polymorphism, and inheritance. Along with this Java also works with three OOPs concept

How Java different to C++ ?

C++ | JAVA

C++ is platform dependent. | Java is platform independent.

C++ uses compiler only. | Java uses compiler and interpreter both.

C++ support pointers and operator overloading. | Java doesn't support pointers and operator overloading concept.

Why Java is so popular programming language?

Java is based on object model hence it is one the popular programming language.

What are the scope of Java Technologies for Web Applications?

Java Technologies for Web Applications are a set of Java-based technologies that are used to develop web applications. These technologies include:

What are the scope of Java Technologies for Web Applications?

- Java Servlet API
- JavaServer Pages
- JavaServer Faces

- Enterprise JavaBeans
- JDBC (Java Database Connectivity)
- Java Messaging Service (JMS):
- JavaMail API:
- JAX-WS

What are the scope of Java Technologies for Web Applications?

Start your Java programming journey today with our Java Programming Online Course, designed for both beginners and advanced learners. With self-paced lessons covering everything from basic syntax to advanced concepts, you'll gain the skills needed to excel in the world of programming. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Track your progress and stay motivated as you master Java. Join now and start your path to becoming a Java expert!

What are the scope of Java Technologies for Web Applications?

- Java
- Tutorials

What are the scope of Java Technologies for Web Applications?

- Java

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers

- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

JavaScript Basics

- Introduction to JavaScript
- JavaScript Versions
- How to Add JavaScript in HTML Document?
- JavaScript Syntax
- JavaScript Output
- JavaScript Comments

JS Variables & Datatypes

- Variables and Datatypes in JavaScript
- Global and Local variables in JavaScript
- JavaScript Let
- JavaScript Const
- JavaScript var

JS Operators

- JavaScript Operators
- Operator precedence in JavaScript
- JavaScript Arithmetic Operators
- JavaScript Assignment Operators
- JavaScript Comparison Operators
- JavaScript Logical Operators
- JavaScript Bitwise Operators
- JavaScript Ternary Operator
- JavaScript Comma Operator
- JavaScript Unary Operators
- JavaScript in and instanceof operators
- JavaScript String Operators

JS Statements

- JavaScript Statements
- JavaScript if-else
- JavaScript switch Statement
- JavaScript Break Statement
- JavaScript Continue Statement
- JavaScript return Statement

JS Loops

- JavaScript Loops
- JavaScript For Loop
- JavaScript While Loop
- JavaScript For In Loop
- JavaScript for...of Loop
- JavaScript do...while Loop

JS Performance & Debugging

- JavaScript | Performance
- Debugging in JavaScript
- JavaScript Errors Throw and Try to Catch

JS Object

- Objects in Javascript
- Introduction to Object Oriented Programming in JavaScript
- JavaScript Objects
- Creating objects in JavaScript
- JavaScript JSON Objects
- JavaScript Object Reference

JS Function

- Functions in JavaScript
- How to write a function in JavaScript ?
- JavaScript Function Call
- Different ways of writing functions in JavaScript
- Difference between Methods and Functions in JavaScript
- Explain the Different Function States in JavaScript
- JavaScript Function Complete Reference

JS Array

- JavaScript Arrays
- JavaScript Array Methods
- Best-Known JavaScript Array Methods
- Important Array Methods of JavaScript
- JavaScript Array Reference

JS String

- JavaScript Strings
- JavaScript String Methods
- JavaScript String Reference

JS Numbers

- JavaScript Numbers
- How numbers are stored in JavaScript ?
- How to create a Number object using JavaScript ?
- JavaScript Number Reference

JS Math

- JavaScript Math Object

- What is the use of Math object in JavaScript ?
- JavaScript Math Reference

JS Map

- Map in JS
- What is JavaScript Map and how to use it ?
- JavaScript Map Reference

JS Set

- Set in JavaScript
- How are elements ordered in a Set in JavaScript ?
- Iterate over Set in JavaScript or JS
- How to sort a set in JavaScript ?
- JavaScript Set Reference

JS Objects

- JavaScript Date
- JavaScript Promise
- JavaScript BigInt
- JavaScript Boolean
- JavaScript Proxy/Handler
- JavaScript WeakMap
- JavaScript WeakSet
- JavaScript Function Generator
- JavaScript JSON

JS Advance

- Arrow functions in JavaScript
- JavaScript this Keyword

- "use strict" in JavaScript
- Introduction to ES6
- JavaScript Hoisting
- Async and Await in JavaScript

JavaScript Exercises

- JavaScript Exercises, Practice Questions and Solutions

JavaScript Tutorial

-
-
-

JavaScript Tutorial

JavaScript is a programming language used to create dynamic content for websites. It is a lightweight, cross-platform, and single-threaded programming language. JavaScript is an interpreted language that executes code line by line providing more flexibility.

JavaScript Tutorial

- HTML adds Structure to a web page, CSS styles it and JavaScript brings it to life by allowing users to interact with elements on the page, such as actions on clicking buttons, filling out forms, and showing animations.
- JavaScript on the client side is directly executed in the user's browser. Almost all browsers have JavaScript Interpreter and do not need to install any software. There is also a browser console where you can test your JavaScript code.
- JavaScript is also used on the Server side (on Web Servers) to access databases, file handling and security features to send responses to browsers.

Hello World Program

This JavaScript Compiler is completely free and easy to use. Here, you can practice various JS Exercises.

Why to learn JavaScript?

- Versatility: JavaScript can be used to develop (usingElectronJS) websites, games (UsingPhaserandThree.js), mobile apps (usingReact Native), and more.
- Client Side:JavaScript is the main language for client-side logic and is supported by almost all browsers. There is a big list of frameworks and libraries likeReact JS,Angular JS, andVue JS.
- Server-Side: With runtime environments likeNode.jsand Frameworks likeExpress.js, JavaScript is now widely used for building server-side applications.
- Machine Learning: With Libraries likeTensorflow.JS,JavaScript can be used to develop and train machine learning models. Please refer toML in JSfor details.

Fundamentals

Let's ExploreJavaScript's fundamentalswhich will help build you a understanding to start with JavaScript

Fundamentals

- Introduction
- Using JS in HTML
- Variables and Datatypes
- Operators
- Control Flow Statements
- Scope
- Browser console
- Quizzes:JavaScript Fundamentals

Functions and Events

Functions in JavaScript are reusable blocks of code that perform a specific task. Events are actions

that happen in the browser, such as mouse clicks, keyboard input, or page loading.

Functions and Events

- Functions
- Function binding
- Hoisting
- Closures
- Higher-Order Functions
- Iterator
- Function Generator
- Events
- Event Loop
- Event Bubbling
- Quizzes: Functions, Event Handling

JavaScript Beginner Projects

Now you have a basic understanding of JavaScript. So start with some beginner level projects to clear your concept and to implement in real world applications.

JavaScript Beginner Projects

- Counter Application
- Prime Number Checker
- Show and Hide Password
- Palindrome Checker App
- JavaScript Carousel
- Email Validator App
- Unicode Character Value
- Random Number Generator

- Random Password Generator

JavaScript JSON

It is a lightweight data format for storing and exchanging data widely used to send data between a server and a client.

JavaScript JSON

- JSON Tutorial
- JSON vs JavaScript Object
- Read JSON File Using JS
- Parse JSON Data in JS
- JavaScript JSON Parser
- JavaScript JSON Complete Reference
- Quizzes:JSON

Object Oriented Programming

Object-Oriented Programming (OOP) in JavaScript, a concept that enables the structure of code by modeling real-world entities as objects with properties and behaviors.

Object Oriented Programming

- Classes
- Class expression
- Object Constructors
- Static Methods
- Prototype
- Constructor Method
- Encapsulation
- Inheritance
- Polymorphism

- Abstraction
- this Keyword
- Quizzes:JavaScript OOP

Inheritance and Prototype Chain

Inheritance allows objects to inherit properties and methods from other objects, enabling code reuse and creating hierarchical relationships between objects.

Inheritance and Prototype Chain

- Prototype
- Prototype Inheritance
- Prototype Chain
- Object Constructors
- Quizzes:Classes and Inheritance

Asynchronous JavaScript

- Callbacks
- Promise
- Promise Chaining
- Async/Await
- Quizzes:Asynchronous JavaScript

Regular Expression and Validation

Regular expressions plays important role for validation. Validations help ensure that data entered by users meets specific criteria.

Regular Expression and Validation

- Regular expressions
- Form Validation

- Email Validation
- Number Validation
- Password Validation
- URL Validation
- UserName Validation
- Quizzes:Regular Expressions

Global Objects

Global objects play an important role in JavaScript development, offering a wide range of functionalities for various tasks

Global Objects

- encodeURIComponent(), encodeURIComponent() and decodeURIComponent()
- eval() Method
- globalThis Property
- Global property
- Infinity
- Undefined vs Undeclared
- Quizzes:Objects

Exception and Error Handling

Exception and Error handling is crucial for ensuring the reliability and stability of JavaScript applications by handling errors effectively

Exception and Error Handling

- Exception Handling
- throw Statement
- try-catch Statement
- Debugging

- Quizzes:Error Handling and Debugging

Testing and Performance Optimization

- Unit testing with Jest
- Memory Management
- Garbage Collection
- Lazy Loading
- Debouncing
- Throttling
- Quizzes:Testing and Optimization

Interesting Facts

This section covers all the interesting facts and features which made JavaScript so popular and will easy if you are switching from other programming languages also

Interesting Facts

- Data Types
- Strings
- Functions
- 'this' keyword
- Set
- Map
- Arrays
- Object
- Complete JavaScript

JavaScript Projects

This section covers some projects to improve and revise your JavaScript Programming Knowledge.

JavaScript Projects

- Button Loading Animation
- Create a Pagination
- Create Popup Box
- Password Generator App
- Word Guessing Game
- Notes App
- Password Strength Checker
- Quiz App with Timer
- Expense Tracker
- Dynamic Resume Builder
- Tap the Geek Game
- Hit the Mouse Game
- Rock Paper and Scissor Game
- Tic-Tac-Toe Game

Practice Quiz

This section offers a collection of practice Quiz designed to test and support understanding of various concepts in JavaScript.

Practice Quiz

- JavaScript Quiz
- Practice Quiz-1
- Practice Quiz-2
- Practice Quiz-3

Interview Questions

This section provides a list of interview questions related to JavaScript.

Interview Questions

- JavaScript Interview Questions and Answers (2025) For Beginners
- Intermediate JavaScript Interview Questions and Answers (2025)
- Advance JS Interview Questions and Answers (2025) For Experienced

Libraries and Frameworks

JavaScript libraries and frameworks play an important role in modern web development. They offer built-in functions and methods that enhance web pages, making them more dynamic and interactive. They handle repetitive tasks, allowing developers to focus on core functionality.

Libraries and Frameworks

Also, they provide project structure and data flow structure that helps to create fast and more reliable applications.

Libraries

Libraries provide pre-built solutions for common tasks. Developers can use these functions instead of writing code from scratch, saving valuable time. Here are a few popular libraries of JavaScript.

Libraries

- Frontend Libraries: React, Preact, Lodash, Moment.js, jQuery, Axios
- Backend Libraries: Socket.io, JWT, Bcrypt, Passport.js, CORS

Frameworks

Frameworks offer a comprehensive structure for building applications. Here are a few popular frameworks of JavaScript.

Frameworks

- Frontend Frameworks: Vue.js, Angular, Next.js, Nuxt.js, Gatsby, Remix
- Backend Frameworks: Express.js, NestJS, Koa.js, Sails.js, Fastify
- FullStack Frameworks: Meteor.js, Next.js, Nuxt.js, RedwoodJS

Frameworks

If you are a person, who is more into study from Courses. JavaScript Course Provides you Complete Beginner to Advanced learning.

What is JavaScript?

JavaScript is the most powerful and versatile web programming language. It is used for making the websites interactive. JavaScript helps us add features like animations, interactive forms and dynamic content to web pages.

What versions of JavaScript does the tutorial cover?

This JS tutorial covers a wide range of JavaScript versions, from the basics to more advanced concepts. It encompasses both older versions like ES5 and ES6, as well as more recent versions and features introduced in ES7, ES8, ES9, and beyond.

What is JavaScript used for?

JavaScript is primarily used for enhancing web pages by adding interactivity, dynamic content, and user-friendly features. It runs directly in web browsers and allows developers to create responsive and engaging user interfaces.

How do I learn JavaScript from scratch?

To learn JavaScript from scratch, start with the basics:

How do I learn JavaScript from scratch?

- Understand variables, data types, and operators.
- Learn about functions, loops, and conditional statements.
- Practice by building small projects and gradually move to more complex ones.

What are JavaScript frameworks?

JavaScript frameworks (such as Angular, or Vue.js) provide a structured way to build web applications. They offer pre-built components, routing, state management, and other tools to

streamline development.

Is this an advanced JavaScript tutorial?

this javascript tutorial covers all important advanced JavaScript concepts such as functional programming, object-oriented programming, asynchronous programming, and more. Beginners can start with the basics and gradually delve into more advanced concepts as they progress through the tutorial.

Is this an advanced JavaScript tutorial?

Master DSA with JavaScript in just 90 days. Explore core DSA concepts, refine your coding skills, and tackle real-world challenges. Take on the Three 90 Challenge? complete 90% of the course in 90 days and earn a 90% refund as a reward for your commitment!

Is this an advanced JavaScript tutorial?

- JavaScript
- Web Technologies
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us

- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS

- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS

- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce

- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams

- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator

- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course

- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Kubernetes

- Introduction to Kubernetes (K8S)
- Kubernetes - Architecture
- Kubernetes - Monolithic Architecture of Kubernetes
- Kubernetes vs Docker

Installation and Setup

- How to Install and Run a Kubernetes Cluster on Ubuntu 22.04 (Step-by-Step)
- How to Install and Configure Kubernetes on Ubuntu?
- How to set up Kubernetes cluster on local machine using minikube ?

Application Deployment

- What are Kubernetes Containers?
- Kubernetes - Introduction to Container Orchestration
- Kubernetes - Images
- Kubernetes - Jobs
- Kubernetes - Labels & Selectors
- Kubernetes - Namespaces
- Kubernetes - Node
- Kubernetes - NodePort Service
- Kubernetes - ClusterIP vs NodePort vs LoadBalancer
- Kubernetes - Services
- Kubernetes Pods: How to Create and Manage Them
- How to Run Shell Commands in Kubernetes Pods or Containers
- Kubernetes - Creating Multiple Container in a Pod
- Kubernetes - Replication Controller
- Kubernetes - Difference Between Replicaset and Replication Controller
- What is Kubernetes Deployment?

Configmaps

- Kubernetes - ConfigMaps
- Kubernetes - Create Config Map From Files
- Kubernetes - Create ConfigMap From YAML File
- Kubernetes - Config Map From Directory
- Kubernetes - Injecting ConfigMap as Files
- Kubernetes - Injecting ConfigMap in Pods

Scaling and Updating Applications

- Kubernetes - Volumes
- Kubernetes - Secrets
- Kubernetes - Working With Secrets
- Kubernetes - Load Balancing Service

Additional Topics

- What is Kubernetes API ?Complete Guide
- Kubernetes - Taint and Toleration
- Kubernetes Resource Model (KRM) and How to Make Use of YAML?
- Installing Private Git Server on K8s Cluster with Gitea and AKS
- Enable Remote Debugging For Java Application Deployed in Kubernetes Environment
- How to Enable JMX For Java Application Running in the Kubernetes Cluster?

Kubernetes Tutorial

-
-
-

Kubernetes Tutorial

In this Kubernetes Tutorial you'll learn all the basics to advanced concepts likeKubernetes

service,Kubernetes cluster construction,Kubernetes deployment,Kubernetes Architecture, etc. This free tutorial on Kubernetes will provide all the essential information needed to understand and work with Kubernetes, including the use of APIs, installation, and Kubernetes cluster construction. Whether you are a beginner or an expert, this tutorial will cover all the necessary details to help you learn and understand Kubernetes.

Kubernetes Tutorial

Kubernetes is a tool that helps us to run and manage applications in containers. It was developed by Google Lab in 2014, and it is also known as k8s. It is an open-source container orchestration platform that automates the deployment, management, and scaling of container-based applications in different kinds of environments like physical, virtual, and cloud-native computing foundations.

Kubernetes Tutorial

For those aiming to deepen their expertise in managing containerized applications and orchestration, consider exploring our [AWS Solutions Architect Certification Training Program](#). This expert-led training complements the knowledge acquired in this tutorial by providing practical, hands-on experience with Kubernetes.

Kubernetes Tutorial

Kubernetes Tutorial

Containers are isolated from each other so that multiple containers can run on the same machine without interrupting anyone else. It allows us to deploy and manage container-based applications across a Kubernetes cluster of machines.

Prerequisites for this Kubernetes Tutorial

Before diving into Kubernetes, it's essential to have a solid understanding of certain concepts and technologies. Here are some prerequisites:

Prerequisites for this Kubernetes Tutorial

- Have good understanding of Container concept & Container Management Tool like Docker or Podman.
- Understanding of distributed system.
- Understanding of REST API
- Basic understanding of YAML

Basics of Kubernetes

Kubernetes Basics will teach you how to manage these ?containers? effectively. Kubernetes is like a ship captain for these containers, organizing and placing them on multiple computers (like a ship carrying many containers). This introduction will explain key Kubernetes ideas like Pods, Services, and Deployments. You?ll learn how Kubernetes keeps your containerized applications running smoothly and efficiently

Basics of Kubernetes

- Introduction to Kubernetes
- Kubernetes ? Installation Methods
- Installation of Kubernetes on Ubuntu
- Kubernetes ? Architecture
- Kubernetes ? Monolithic Architecture of Kubernetes
- Kubernetes vs Docker
- Kubernetes ? Concept of Containers
- Kubernetes ? Introduction to Container Orchestration
- Kubernetes ? Images
- Kubernetes ? Jobs
- Kubernetes ? Labels & Selectors
- Kubernetes ? Namespace
- Kubernetes ? Node
- Kubernetes ? Node Port Service

- Kubernetes ? Cluster IP vs Node-Port
- Kubernetes ? Service
- Kubernetes ? Service DNS
- Kubernetes ? Pod
- Kubernetes ? Run a Command in Pod's Containers
- Kubernetes ? Create Multiple Container in a Pod
- Kubernetes ? Replication Controller
- Kubernetes ? Difference Between Replicaset and Replication Controller
- Kubernetes ? Deployments
- Kubernetes ? Volumes
- Kubernetes ? Secrets
- Kubernetes ? Working with Secrets
- How to set up a Kubernetes cluster on a local machine using minikube?
- Kubernetes ? Physical Servers vs Virtual Machines vs Containers

Advanced Kubernetes

In this Advanced Kubernetes section we will understand how to manage complex containerized applications. This section assumes you're familiar with Kubernetes basics and explores powerful features for scaling, security, and automation. We'll delve into concepts like deployments with rollbacks and health checks, advanced networking configurations for service communication, and tools for managing storage and persistent data for your containerized applications.

Advanced Kubernetes

- Kubernetes ? API
- Kubernetes ? Taint and Toleration
- Kubernetes ? Kubectl
- Kubernetes ? Kubectl Commands
- Kubernetes ? Kubectl Delete

- Kubernetes ? Load Balancing Service
- Kubernetes ? Kubectl Create and Kubectl Apply
- Kubernetes ? ConfigMap
- Kubernetes ? Create Config Map From Files
- Kubernetes ? Create ConfigMap from YAML
- Kubernetes ? ConfigMap from Directories
- Kubernetes ? Injecting ConfigMap as Files
- Kubernetes ? Injecting ConfigMap in Pods
- Kubernetes Resource Model (KRM) and How to Make Use of YAML?
- Installing Private Git Server on K8s Cluster with Gitea and AKS
- Enable Remote Debugging For Java Application Deployed in Kubernetes Environment
- How to Enable JMX For Java Applications Running in the Kubernetes Cluster?

Why Do We Need Kubernetes?

There are several reasons to learn Kubernetes like easy scaling of applications, self-healing, portability, and automation. It is very helpful for running microservices and distributed systems.

Why Do We Need Kubernetes?

For example: You have a couple of applications to deploy so, you can package it into a container and run it on a server containing a Docker engine or any other container engine. You package the application into a container using a Docker file and host it on a port for the external world to access it.

Why Do We Need Kubernetes?

But there is a drawback is that it is only running on a single server so, if at that point any failure occurs it becomes an application failure, to handle the single point of failure google introduced Kubernetes to scale applications.

Kubernetes Architecture

Below you will find the image that describe the architecture of Kubernetes.

Kubernetes Architecture

Kubernetes Architecture

Kubernetes Architecture

Features of Kubernetes

- Consistent Development, management, and deployment
- Container-based infrastructure
- Utilization of resources in higher density
- Each component is like a separate unit
- Application-centric infrastructure
- Auto scalability
- Consistency is maintained across testing and development

Advantages of Kubernetes

- Container Orchestration:Kubernetes automates the deployment, scaling, and management of containerized applications, ensuring efficient resource utilization and seamless deployment updates.
- Scalability:Kubernetes enables horizontal scaling of applications by adding or removing container replicas based on demand, ensuring optimal performance during peak loads.
- High Availability:With automated health checks and self-healing capabilities, Kubernetes ensures that applications remain available and responsive, reducing downtime and enhancing reliability.
- Fault Tolerance:Kubernetes manages application failures by automatically restarting containers or shifting traffic to healthy instances, improving overall application resilience.
- Flexibility:Supports multi-cloud and hybrid-cloud environments, allowing deployment across various infrastructure providers and on-premises data centers without vendor lock-in.
- Resource Efficiency:Efficiently manages computing resources like CPU and memory, optimizing

utilization and reducing costs by scaling resources based on application demands.

- Automated Operations: Simplifies complex operational tasks such as load balancing, storage orchestration, and networking configuration through declarative APIs and automation.

Conclusion

This tutorial provided a comprehensive overview of Kubernetes, including its history, key features, and how it can be used to manage and deliver containerized applications. We covered the use of Kubernetes APIs, installation, and cluster construction. Whether you are new to Kubernetes or an experienced user, this tutorial will provide you with the information you need to understand and work with this powerful open-source platform. Kubernetes is widely used in the industry and continues to be a popular choice for managing containerized applications in a production environment. Keep learning and experimenting with Kubernetes to discover its full potential.

How does Kubernetes work?

Kubernetes works by allowing users to define how their applications should run and then automatically handling the deployment, scaling, and management of those applications based on the defined specifications.

What are the key benefits of using Kubernetes?

Some key benefits of using Kubernetes include increased operational efficiency, improved resource utilization, automatic scaling of applications, and enhanced portability across various infrastructure environments.

Is Kubernetes suitable for small businesses?

Yes, Kubernetes can be beneficial for small businesses as it provides a scalable and efficient way to manage containerized applications, enabling them to run seamlessly regardless of the scale.

Are there any alternatives to Kubernetes?

Yes, there are other container orchestration platforms like Docker Swarm and Apache Mesos, but

Kubernetes is one of the most popular and widely adopted solutions due to its robust feature set and active community support.

Is Kubernetes a Docker?

Docker is a container runtime, Kubernetes is a platform for running and managing containers from many container runtimes.

Is Kubernetes a Docker?

Is Kubernetes a Docker?

Take your DevOps skills to the next level with our DevOps Engineering - Planning to Production course. From understanding the core principles of DevOps to implementing continuous integration and delivery pipelines, this course provides a comprehensive, self-paced learning experience. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Stay motivated, track your progress, and challenge yourself to become a certified DevOps expert. Join now and start your journey toward mastering DevOps engineering!

Is Kubernetes a Docker?

A

Is Kubernetes a Docker?

- DevOps
- Kubernetes
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us

- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL

- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML

- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science

- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies

- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor

- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

MySQL Basics

- What is MySQL?
- MySQL DATE Data Type
- How to Install MySQL on Windows?
- How to Install MySQL on Linux?
- How to Install MySQL on macOS?
- How to Install MySQL on Fedora?
- How to Install SQL Workbench For MySQL on Windows?
- How to Install MySQL WorkBench on Ubuntu?
- How to Install SQL Workbench For MySQL on Linux?
- Connecting to MySQL Using Command Options
- Java Database Connectivity with MySQL
- Connect MySQL database using MySQL-Connector Python
- How to make a connection with MySQL server using PHP ?
- How to Connect to Mysql Server Using VS Code and Fix errors?
- How to Connect Node.js Application to MySQL ?

MySQL User Management

- MySQL CREATE USER Statement
- MySQL | DROP USER
- MySQL | USER() Function
- MySQL | Change User Password

MySQL Managing Databases

- MySQL Create Database Statement
- MySQL | Common MySQL Queries
- MySQL | Common MySQL Queries
- MySQL Drop Database

- Python MySQL - Create Database
- NodeJS MySQL Create Database

MySQL Managing Tables

- MySQL CREATE TABLE
- MySQL | Common MySQL Queries
- MySQL RENAME TABLE Statement
- Drop Multiple Tables in MySQL
- MySQL Temporary Table
- Drop Multiple Tables in MySQL
- Drop Multiple Tables in MySQL
- Drop Multiple Tables in MySQL
- Node.js MySQL Drop Table
- Inserting data into a new column of an already existing table in MySQL using Python
- MySQL | Common MySQL Queries
- Python: MySQL Create Table
- Python: MySQL Create Table
- PHP | MySQL (Creating Table)
- Node.js MySQL Create Table
- Create Table From CSV in MySQL
- Node.js MySQL Drop Table
- Python MySQL - Drop Table
- How to Rename a MySQL Table in Python?

MySQL Query

- MySQL | Common MySQL Queries
- Nested Select Statement in MySQL
- MySQL DISTINCT Clause

- INSERT() function in MySQL
- MySQL Derived Table
- MySQL Insert Multiple Rows
- MySQL INSERT INTO SELECT Statement
- MySQL INSERT ON DUPLICATE KEY UPDATE Statement
- MySQL Insert Date Time
- MySQL Insert Date Time
- MySQL UPDATE Statement
- MySQL DELETE Statement
- How to Delete Duplicate Rows in MySQL?
- MySQL DELETE JOIN
- MySQL - ON DELETE CASCADE Constraint
- Truncate All Tables in MySQL
- PHP | Inserting into MySQL database
- Python MySQL - Update Query
- PHP | MySQL UPDATE Query
- Node.js MySQL Update Statement

MySQL Clauses

- MySQL WHERE Clause
- MySQL ORDER BY Clause
- MySQL | PARTITION BY Clause
- Queries using AND ,OR ,NOT operators in MySQL
- Queries using AND ,OR ,NOT operators in MySQL
- MySQL EXISTS Operator

MySQL Aggregate Functions

- COUNT() Function in MySQL

- SUM() Function in MySQL
- AVG() Function in MySQL

MySQL Data Constraints

- MySQL NOT NULL Constraint
- MySQL UNIQUE Constraint
- MySQL Primary Key
- MySQL FOREIGN KEY Constraint
- MySQL COMPOSITE KEY
- MySQL UNIQUE Constraint
- MySQL DEFAULT Constraint

MySQL Joining Data

- MySQL Inner Join
- MySQL LEFT JOIN
- MySQL RIGHT JOIN
- MySQL SELF JOIN
- MySQL CROSS JOIN
- MySQL UPDATE JOIN
- MySQL DELETE JOIN
- MySQL | Recursive CTE (Common Table Expressions)

MySQL Functions

- DATE() in MySQL
- TRUNCATE() Function in MySQL
- Mathematical functions in MySQL
- MySQL | CONVERT() Function
- LTRIM() Function in MySQL

- UCASE() or UPPER() Function in MySQL
- RTRIM() Function in MySQL
- MySQL ISNULL() Function
- IFNULL in MySQL
- MySQL CASE() Function
- MySQL | CAST() Function

MySQL Indexes

- MySQL CREATE INDEX Statement
- MySQL DROP INDEX Statement
- MySQL Unique Index
- Difference Between Clustered and Non-Clustered Index

MySQL Miscellaneous Topics

- Different types of Procedures in MySQL
- MySQL Vulnerabilities
- MySQL | Common MySQL Queries
- MySQL Interview Questions

MySQL Tutorial

-
-
-

MySQL Tutorial

This MySQL Tutorial is made for both beginners and experienced professionals. Whether you're starting with MySQL basics or diving into advanced concepts, this free tutorial is the ideal guide to help you learn and understand MySQL, no matter your skill level.

MySQL Tutorial

From setting up your database to performing complex queries and database administration tasks, we'll guide you through each step with clear explanations and practical examples. Dive into our MySQL tutorial and discover how to manage and optimize your databases efficiently, ensuring robust data management for your applications.

What is a Database?

A database is a digital system designed for the storage and arrangement of data. Think of it as an online filing system that allows you to store and quickly access a vast amount of information. Databases facilitate the efficient management of data, enabling the simple addition, modification, removal, and access of information. They serve numerous uses such as websites, applications, and enterprises to manage extensive data in an organized and secure manner.

What is MySQL?

MySQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) to manage data. Developed by MySQL AB and now owned by Oracle Corporation, it's widely used due to its reliability, speed, and ease of use. MySQL is a key component in many web applications, forming the backbone of popular websites and services.

What is MySQL?

It allows users to create, modify, and maintain databases, supporting operations like data insertion, querying, updating, and deletion. Ideal for both small and large-scale applications, MySQL powers various types of systems, from personal projects to complex enterprise environments.

What is MySQL?

MySQL is a Relational Database Management System (RDBMS) software that provides many features, which are as follows:

What is MySQL?

- Data Storage: Efficiently stores large amounts of data.
- Data Retrieval: Allows quick and easy access to data.
- Data Manipulation: Supports operations like inserting, updating, and deleting data.
- Data Security: Offers robust security features to protect data.
- Scalability: Can handle small to large applications with ease.

How MySQL Works?

MySQL works like this:

How MySQL Works?

- Client Request: You send a request to the MySQL server using an application or a command-line tool.
- Connection: The server connects with your application to start a session.
- SQL Parsing: The server checks your SQL query for any errors.
- Query Optimization: The server figures out the best way to execute your query efficiently.
- Execution: The server runs the query. It reads or writes data to the database as needed.
- Storage Engine: This is where data is stored on the disk. MySQL uses different storage engines like InnoDB or MyISAM for this.
- Result Generation: The server creates the result based on your query.
- Response: The server sends the results back to your application.
- Client Interaction: Your application displays the data to you.
- Transaction Management: For complex operations, MySQL ensures everything runs smoothly and data remains accurate.
- Logging and Recovery: MySQL keeps logs to help recover data if something goes wrong.
- Replication and Backup: MySQL can copy data to other servers for safety and better performance.

It also supports data backups to keep your information safe.

How MySQL Works?

In simple terms, MySQL receives your request, processes it efficiently, interacts with stored data, and returns the results to you, all while ensuring data safety and reliability.

How MySQL Works?

In this tutorial, we will learn about:

How MySQL Works?

Table of Content

How MySQL Works?

- What is MySQL?
- How MySQL Works?
- History of MySQL
- Prerequisites for this MySQL Tutorial
- MySQL Basics
- MySQL User Management
- MySQL Managing Databases
- MySQL Managing Tables
- MySQL Query
- MySQL Clauses
- MySQL Operators
- MySQL Aggregate Functions
- MySQL Data Constraints
- MySQL Joining Data
- MySQL Functions
- MySQL Views
- MySQL Indexes

- MySQL Triggers
- MySQL Miscellaneous Topics
- Features of MySQL
- MySQL Jobs and Opportunities
- MySQL Online Interview Questions

Prerequisites for this MySQL Tutorial

- Basic understanding of databases
- Knowledge of SQL
- Familiarity with programming concepts
- Command-line skills
- Grasp of relational database concepts
- Awareness of data types
- Text editor or IDE proficiency

MySQL Basics

MySQL is a widely used open-source RDBMS. Now, in this section of the MySQL tutorial, we will provide you with all the basic concepts of MySQL so that you can easily manage MySQL effectively.

MySQL Basics

- What Is MySQL?
- MySQL Data Types
- How to Install MySQL Database Server using Windows
- How to Install MySQL Database Server using Linux
- How to Install MySQL Database Server using MacOS
- How to Install MySQL Database Server using Fedora
- MySQL Workbench on Windows
- MySQL Workbench on Ubuntu

- MySQL Workbench on Linux
- How to Connect to MySQL Server using command options
- How to Connect to MySQL Server using Java
- How to Connect to MySQL Server python connector
- How to Connect to MySQL Server Using PHP
- How to Connect to MySQL Server using VS code
- How to Connect to MySQL Server using Nodejs

MySQL User Management

MySQL user management encompasses the creation, modification, and deletion of user accounts, along with controlling their access privileges to databases and objects within the MySQL server. In this section, we will explore how to use these queries.

MySQL User Management

- MySQL Create User
- MySQL Drop User
- MySQL Show User
- Change User Password

MySQL Managing Databases

MySQL offers a robust system for managing databases, allowing users to create, modify, and interact with data efficiently. In this section of the MySQL tutorial, you will explore how to create a database, drop a database, and other queries to manage the database.

MySQL Managing Databases

- Create Databases
- Select a Databases
- Show databases
- Drop Databases

- MySQL Create Databases using python
- MySQL Create Databases using Node.js

MySQL Managing Tables

MySQL provides a comprehensive set of functionalities for managing individual tables within a database. These functionalities encompass creating tables with specific data structures, including defining columns, data types, and constraints. Explore this section to get a better understanding of MySQL Table.

MySQL Managing Tables

- Mysql Create Tables
- Mysql DESCRIBE Table
- Mysql Rename Tables
- Mysql Drop Tables
- Mysql Temporary Tables
- Delete multiple tables
- How to Repair MySQL Tables
- Mysql DROP Multiple TABLE
- Mysql DROP TABLE in python
- Mysql DROP TABLE in Node.js
- Mysql Add Columns
- Mysql Drop Columns
- MySQL AUTO_INCREMENT
- MySQL Create table using python
- Mysql Create table using php
- Mysql Create table using Node.js
- Mysql Create table from CSV
- Mysql Drop Table using Node.js

- Mysql Drop Table using Python
- Rename Table using Python

MySQL Query

MySQL queries are the heart of interacting with your database. They act as instructions, retrieving specific data, filtering results, and performing calculations.

MySQL Query

- MySQL Common Queries
- MySQL Nested SELECT Statement
- MySQL SELECT Statement
- MySQL SELECT DISTINCT
- MySQL INSERT Statement
- MySQL Derived Tables
- MySQL Insert Multiple Rows
- MySQL INSERT INTO SELECT
- MySQL Insert On Duplicate Key Update
- MySQL INSERT IGNORE
- MySQL Insert DateTimes
- MySQL Insert Dates
- MySQL UPDATE Statement
- MySQL DELETE Statement
- MySQL DELETE Duplicate Rows
- MySQL DELETE JOIN
- MySQL ON DELETE CASCADE
- MySQL TRUNCATE TABLE
- Insert Multiple Rows mysql using PHP
- Python Mysql Update statement

- PHP Mysql Update statement
- Node.js Mysql Update statement

MySQL Clauses

MySQL clauses are the building blocks of powerful database queries. They act like instructions within a statement, specifying actions like filtering data, sorting results, and grouping information

MySQL Clauses

- MySQL WHERE Clause
- MySQL HAVING Clause
- MySQL ORDER By Clause
- MySQL Group By Clause
- MySQL LIMIT Clause
- MySQL PARTITION BY

MySQL Operators

MySQL operators are the building blocks for manipulating data within your queries. They perform various functions like comparisons, logical operations, and arithmetic calculations, allowing you to precisely control how data is filtered, transformed, and retrieved. Understanding these operators is essential for crafting effective and efficient MySQL queries.

MySQL Operators

- MySQL AND Operator,
- MySQL LIKE Operator
- MySQL IN Operator
- MySQL NOT Operator
- MySQL NOT EQUAL Operator
- MySQL IS NULL Operator
- MySQL UNION Operator

- MySQL UNION ALL Operator
- MySQL EXCEPT Operator
- MySQL BETWEEN Operator
- MySQL ALL, ANY Operator
- MySQL INTERSECT Operator
- MySQL EXISTS Operator
- MySQL CASE Operator

MySQL Aggregate Functions

In MySQL, aggregate functions condense large datasets into a single, meaningful value. They perform calculations like finding the average salary, counting the number of customers, or identifying the highest product price.

MySQL Aggregate Functions

- MySQL Aggregate Function
- MySQL Count() Function
- MySQL SUM() Function
- MySQL MIN() Function
- MySQL MAX() Function
- MySQL AVG() Function

MySQL Data Constraints

MySQL helps keep your data accurate by setting rules for what can be stored. These rules, called constraints, prevent invalid entries like missing values or duplicates, ensuring clean and reliable data.

MySQL Data Constraints

- MySQL NOT NULL
- MySQL UNIQUE

- MySQL Primary Key
- MySQL Foreign Key
- MySQL Composite Key
- MySQL Unique Key
- MySQL Alternate Key
- MySQL CHECK Cons
- MySQL DEFAULT Constraint

MySQL Joining Data

Need to combine data from multiple tables? MySQL's JOINS come to the rescue! They merge rows based on matching values, letting you see connections and gain insights you wouldn't get from separate tables.

MySQL Joining Data

- MySQL JOIN
- MySQL Outer Join
- MySQL Inner Join
- MySQL Left Join
- MySQL Right Join
- MySQL Self Join
- MySQL Full Join
- MySQL Cross Join
- MySQL UPDATE JOIN
- MySQL DELETE JOIN
- MySQL Recursive Join

MySQL Functions

MySQL provides a vast library of functions, acting as powerful tools to manipulate and analyze your

data. These functions can perform various tasks, including mathematical calculations, string manipulation, date and time operations, and data aggregation. By leveraging these functions, you can efficiently transform and analyze your data, extracting valuable insights.

MySQL Functions

- MySQL Date Functions
- MySQL String Functions
- MySQL TRUNCATE Function
- MySQL Window Functions
- MySQL Math Functions
- MySQL Statistical Functions
- MySQL JSON Functions
- MySQL Convert Functions
- MySQL Datatype Functions
- MySQL LTRIM Function
- MySQL UPPER Function
- MySQL RTRIM Function
- MySQL ISNULL Function
- MySQL IFNULL Function
- MySQL CASE Function
- MySQL CAST Function

MySQL Views

Views act like virtual tables, simplifying complex queries. Think of them as pre-written queries you can reference easily, like a shortcut. This saves time and improves code readability, making your database interactions more efficient.

MySQL Views

- MYSQL VIEW
- MySQL CREATE VIEW
- MySQL DROP VIEW
- MySQL UPDATE VIEW
- MySQL RENAME VIEW

MySQL Indexes

Imagine flipping through a giant phonebook without an index. Indexes in MySQL act like helpful tools, speeding up data searches. By organizing data efficiently, they help retrieve specific information quickly, making your queries run like a charm.

MySQL Indexes

- MySQL Indexes
- MySQL Create Index
- MySQL Drop Index
- MySQL Show Indexes
- MySQL Unique Index
- Clustered Index vs Non-Clustered Index

MySQL Triggers

MySQL triggers are mini-programs that run automatically in response to specific events like data insertion, updates, or deletion. They can enforce data integrity, maintain logs, or even perform calculations, acting as silent guardians of your database.

MySQL Triggers

- MySQL Trigger
- MySQL Create Trigger
- MySQL Show Trigger
- MySQL DROP Trigger

- MySQL Before Insert Trigger
- MySQL After Insert Trigger
- MySQL BEFORE UPDATE Trigger
- MySQL AFTER UPDATE Trigger
- MySQL BEFORE DELETE Trigger
- MySQL AFTER DELETE Trigger

MySQL Miscellaneous Topics

The "MySQL Miscellaneous Topic" section delves into lesser-known functionalities or advanced techniques. This might include specific functions, managing special data types, or troubleshooting uncommon issues.

MySQL Miscellaneous Topics

- MySQL Subqueries
- MySQL Procedure
- MySQL Transaction
- MySQL Security
- MySQL Vulnerabilities
- MySQL Partition
- MySQL SQL Injection
- MySQL Common Table Expressions
- MySQL Cursors
- MySQL Common Queries
- MySQL Interview Questions
- MySQL Handling NULL Values

Features of MySQL

MySQL simplifies data management by providing a user-friendly platform for efficient storage,

retrieval, and organization. It ensures robust security, accommodating multiple users and transactions seamlessly. Commonly used for websites and applications, MySQL enhances data handling. Its features include simplicity in querying, scalability for varying data needs, and compatibility with various programming languages. Overall, MySQL's versatility and accessibility make it a reliable choice for users looking to manage and interact with their data effectively.

History of MySQL

MySQL is a popular open-source relational database management system. Here's a concise history:

History of MySQL

- 1995: The MySQL database was founded by Michael Widenius, David Axmark, and Allan Larsson. It was initially designed to be a budget-friendly option compared to pricier databases and its foundation was in SQL (Structured Query Language).
- 2000: MySQL AB, the company behind MySQL, released the database under the GNU General Public License (GPL), making it free to use and modify.
- 2001-2005: MySQL gained popularity due to its reliability, performance, and ease of use, becoming a preferred database for web applications. During this time, several major releases added features like subqueries, views, and stored procedures.
- 2008: Sun Microsystems acquired MySQL AB for \$1 billion, integrating MySQL into its suite of software products.
- 2010: Oracle Corporation acquired Sun Microsystems, and with it, MySQL. This acquisition raised concerns in the open-source community about the future of MySQL, leading to the creation of MariaDB, a fork of MySQL led by Monty Widenius.
- 2010s: MySQL continued to evolve under Oracle's stewardship, with significant updates improving performance, scalability, and security. Oracle also maintained dual licensing for MySQL, offering both open-source and commercial versions.
- Present: MySQL remains widely used across various industries, particularly for web applications. It

continues to compete with other database systems and is a critical component of the LAMP (Linux, Apache, MySQL, PHP/Perl/Python) stack.

MySQL Jobs and Opportunities

There are numerous companies around the globe seeking MYSQL professionals, and they pay high packages. The average salary of MYSQL developers is around 40,000 to 65,000 INR. In this section, we have listed some of the top giant companies that hire MYSQL experts.

MySQL Jobs and Opportunities

- Google
- Microsoft
- Amazon
- Meta
- Apple
- Accenture
- Deloitte
- McKinsey & Company
- KPMG
- JPMorgan Chase
- Bank of America
- HSBC
- Netflix
- Capgemini
- Wipro
- Infosys
- Tata Consultancy

MySQL Online Interview Questions

Conclusion

In conclusion, this MYSQL tutorial caters to both beginners and professionals, guiding you from basics to advanced topics. With prerequisites like database understanding and SQL knowledge, it covers MYSQL essentials, user management, database and table handling, and common queries. Master MYSQL for effective data management!

How to use MySQL step by step?

Follow the steps: Install MySQL, create databases, manage tables, and execute queries using MySQL commands or tools like MySQL Workbench.

Is MySQL easy for beginners?

Yes, MySQL is beginner-friendly with clear syntax and documentation, making it accessible for those new to databases.

Is SQL and MySQL the same?

No, SQL is a language used for managing databases, while MySQL is a specific relational database management system that utilizes SQL for querying and managing data.

Is SQL and MySQL the same?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Is SQL and MySQL the same?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Is SQL and MySQL the same?

Is SQL and MySQL the same?

- MySQL
- Databases

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation

- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp

- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux

- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures

- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Node.js Basic

- Node.js Introduction
- How to Install Node.js on Linux
- How to Install Node.js on Windows
- Node.js Basics
- Node First Application
- Node.js REPL (READ, EVAL, PRINT, LOOP)
- NodeJS NPM
- Node.js Global Objects
- Node.js Modules
- Node.js Local Module

Node.js Assert Module

- Node.js Assert Module
- Node.js assert() Function
- Node.js assert.deepStrictEqual() Function
- Node.js assert.doesNotThrow() Function
- Node.js assert.equal() Function
- Node.js assert.ifError() Function
- Node.js assert.match() Function
- Node.js assert.notDeepEqual() Function
- Node.js Assert Complete Reference

Node.js Buffer Module

- Node.js Buffers
- Node.js Buffer.copy() Method
- Node.js Buffer.includes() Method
- Node.js Buffer.compare() Method

- Node.js Buffer.alloc() Method
- Node.js Buffer.equals() Method
- Node.js Buffer.subarray() Method
- Node.js Buffer.readIntBE() Method
- Node.js Buffer.write() Method
- Node.js Buffer Complete Reference

Node.js Console Module

- Node.js Console
- Node.js console.assert() Method
- Node.js console.clear() Method
- Node.js console.count() Method
- Node.js console.countReset() Method
- Node.js console.debug() Method
- Node.js console.dir() Method
- Node.js console.error() Function
- Node.js console.info() Method
- Node.js Console Complete Reference

Node.js Crypto Module

- Node.js cipher.final() Method
- Node.js cipher.update() Method
- Node.js crypto.getCiphers() Method
- Node.js crypto.createDecipheriv() Method
- Node.js crypto.createCipheriv() Method
- Node.js crypto.getDiffieHellman() Method
- Node.js crypto.pbkdf2() Method
- Node.js crypto.createHash() Method

- Node.js crypto.createHmac() Method
- Node.js Crypto Complete Reference

Node.js DNS Module

- Node.js DNS
- Node.js dns.getServers() Method
- Node.js dns.lookup() Method
- Node.js dns.lookupService() Method
- Node.js dns.resolve() Method
- Node.js dns.resolve4() Method
- Node.js dns.resolve6() Method
- Node.js dns.resolveAny() Method
- Node.js dns.resolveCname() Method
- Node.js DNS Complete Reference

Node.js File System Module

- Node.js File System
- Node JS fs.readFile() Method
- Node.js fs.exists() Method
- Node fs.existsSync() Method
- Node fs.mkdir() Method
- Node.js fs.truncate() Method
- Node.js fs.renameSync() Method
- Node.js fs.rmdir() Method
- Node.js fs.stat() Method
- Node.js File System Complete Reference

Node.js Globals

- Node.js Timers module
- Import and Export in Node.js
- Node Export Module
- Node.js URL() Method
- Node.js URLSearchParams API
- Node.js Globals Complete Reference

Node.js HTTP Module

- Node.js HTTP Module
- Node.js new Agent() Method
- Node.js agent.createConnection() Method
- Node.js agent.maxSockets Method
- Node.js agent.maxFreeSockets Method
- Node.js http.ClientRequest.abort() Method
- Node.js http.ClientRequest.connection Property
- Node.js http.ClientRequest.protocol Method
- Node.js http.ClientRequest.aborted Property
- Node.js HTTP Module Complete Reference

Node.js HTTP2 Module

- Node.js Http2Session Close Event
- Node.js http2session.state Method
- Node.js Http2Session timeout Event
- Node.js http2session.socket Method
- Node.js http2session.remoteSettings Method
- Node.js http2session.destroyed Method
- Node.js http2session.type Method
- Node.js http2session.encrypted Method

- Node.js http2session.localSettings Method
- Node.js HTTP2 Complete Reference

Node.js OS Module

- Node.js OS
- Node.js os.EOL
- Node.js os.arch() Method
- Node.js os.cpus() Method
- Node.js os.endianness() Method
- Node.js os.freemem() Method
- Node.js os.getPriority() Method
- Node.js os.homedir() Method
- Node.js os.hostname() Method
- Node.js OS Complete Reference

Node.js Path Module

- Node.js path.basename() Method
- Node.js path.delimiter Property
- Node.js path.dirname() Method
- Node.js path.extname() Method
- Node.js path.format() Method
- Node.js path.isAbsolute() Method
- Node.js path.join() Method
- Node.js path.normalize() Method
- Node.js path.parse() Method
- Node.js Path Module Complete Reference

Node.js Process Module

- Node.js process.arch Property
- Node.js process.argv Property
- Node.js process.argv0 Property
- Node.js process.chdir() Method
- Node.js process.config Property
- Node.js process.cpuUsage() Method
- Node process.cwd() Method
- Node.js process.debugPort Property
- Node.js process.env Property
- Node.js Process Complete Reference

Node.js Query String Module

- Node.js querystring.parse() Method
- Node.js querystring.stringify() Method
- Node.js Query String Complete Reference

Node.js Stream Module

- Node.js Streams
- Node.js Stream writable.cork() Method
- Node.js Stream writable.writableLength Property
- Node.js Stream writable.writableObjectMode Property
- Node.js Stream writable.writableFinished Property
- Node.js Stream writable.destroy() Method
- Node.js Stream writable.writableCorked Property
- Node.js Stream writable.end() Method
- Node.js stream.Writable close Event
- Node.js Stream Complete Reference

Node.js String Decoder Module

- Node.js stringDecoder.end() Method
- Node.js stringDecoder.write() Method
- Node.js String Decoder Complete Reference

Node.js Timers Module

- Node.js Immediate Timer Class
- Node.js Timeout Timer Class
- Node.js Timers Complete Reference

Node.js TLS/SSL Module

- Node.js tls.createServer() Method
- Node.js tls.rootCertificates Property
- Node.js tls.getCiphers() Method
- Node.js request.writableEnded Property
- Node.js tlsSocket.address() Method
- Node.js tlsSocket.authorized Property
- Node.js tlsSocket.authorizationError Property
- Node.js tlsSocket.disableRenegotiation() Method
- Node.js TLS/SSL Complete Reference

Node.js UDP/DataGram Module

- Node.js socket.address() Method
- Node.js socket.addMembership() Method
- Node.js socket.bind() Method
- Node.js socket.connect() Method
- Node.js socket.dropMembership() Method
- Node.js socket.getSendBufferSize() Method

- Node.js socket.getRecvBufferSize() Method
- Node.js socket.setTTL() Method
- Node.js socket.setMulticastTTL() Method
- Node.js UDP/DataGram Complete Reference

Node.js URL Module

- Node.js URL.hash API
- Node.js URL.host API
- Node.js URL.origin API
- Node.js URL.username API
- Node.js URL.search API
- Node.js URL.port API
- Node.js URL.pathname API
- Node.js URL.password API
- Node.js URL Complete Reference

Node.js Utility Module

- Node.js Utility Module
- Node.js util.callbackify() Method
- Node.js util.debuglog() Method
- Node.js util.format() Method
- Node.js util.inherits() Method
- Node.js util.formatWithOptions() Method
- Node.js util.inspect() Method
- Node util.promisify() Method
- Node.js util.isDeepStrictEqual() Method
- Node.js Utility Complete Reference

Node.js V8 Module

- Node.js v8.cachedDataVersionTag() Method
- Node.js v8.getHeapSpaceStatistics() Method
- Node.js v8.getHeapStatistics() Method
- Node.js v8.serialize() Method
- Node.js v8.deserialize() Method
- Node.js v8.Serializer.writeHeader() Method
- Node.js v8.Serializer.writeValue() Method
- Node.js v8.Serializer.releaseBuffer() Method
- Node.js v8.Serializer.writeUint32() Method
- Node.js V8 Complete Reference

Node.js VM Module

- Node.js Constructor: new vm.Script() Method
- Node.js script.createCachedData() Method
- Node.js script.runInContext() Method
- Node.js script.runInNewContext() Method
- Node.js | script.runInThisContext() Method
- Node.js vm.createContext() Method
- Node.js vm.runInThisContext() Method
- Node.js vm.isContext() Method
- Node.js vm.runInContext() Method
- Node.js VM Complete Reference

Node.js Zlib Module

- Node.js zlib.constants Property
- Node.js zlib.createBrotliCompress() Method
- Node.js zlib.createBrotliDecompress() Method

- Node.js zlib.createUnzip() Method
- Node.js zlib.createDeflateRaw() Method
- Node.js zlib.createGunzip() Method
- Node.js zlib.createInflateRaw() Method
- Node.js zlib.createDeflate() Method
- Node.js zlib.createInflate() Method
- Node.js Zlib Complete Reference

Node.js Questions

- Why Node.js ?
- How to change Node console font color ?
- How to Read Command Line Arguments in Node ?
- What is the difference between --save and --save-dev in Node.js ?
- Essence of Node.js
- Reading Environment Variables From Node.js
- How to install the previous version of Node and npm?
- How to Create a Directory using Node.js ?
- Node.js Questions Complete Reference
- Node.js Examples

Node.js Tutorial

-
-
-

Node.js Tutorial

Node.js is a powerful, open-source, and cross-platform JavaScript runtime environment built on Chrome's V8 engine.

Node.js Tutorial

- It allows you to run JavaScript code outside the browser, making it ideal for building scalable server-side and networking applications.
- JavaScript was earlier mainly used for frontend development. With Node JS (Introduced in 2009), JavaScript became a backend language as well.
- Non-blocking, event-driven architecture for high performance.
- Supports the creation of REST APIs, real-time applications, and microservices.
- Comes with a rich library of modules through npm (Node Package Manager).

Node.js Tutorial

To start with Node.js, you need to install and set it up on your machine. Follow these articles to install depending on your system:

Node.js Tutorial

- How to Install Node.js on Windows?
- Installation of Node JS on Linux
- How to Install NodeJS on MacOS

Node.js Tutorial

Let us now take a look at our first code example.

Node.js Tutorial

It will start a server, and when you visit <http://localhost:3000>, it will display

Node.js Tutorial

In this example

Node.js Tutorial

- The `http` module is imported to create a basic HTTP server.
- The `createServer()` method is used to handle incoming requests and send responses.

- The server listens on port 3000, and a message is displayed in the browser when accessed.

Why Learn Node.js

- Enables the use of JavaScript for both frontend and backend development.
- Supports building real-time applications like chat apps and gaming servers.
- Provides high scalability for I/O-heavy applications.
- Backed by a vibrant community and extensive library support.

Getting Started with Node.js Tutorial

Prerequisites: JavaScript and basic knowledge of web development

Basics of Node.js

- Node.js Introduction
- Why Node.js?
- Node.js NPM (Node.js Package Manager)
- Installation of Node.js on Linux
- Installation of Node.js on Windows
- Node.js Basics
- Node.js First Application
- Node.js REPL
- Node.js modules
- Node.js Start and Run Server
- Node.js Blocking and Non-Blocking
- Node.js Callback Concept
- Node.js Debugging
- Node.js Set Console Font Color
- Node.js Web Server
- Node.js Create and Publish NPM packages

- Node.js ?save and ?save-dev
- Node.js Event Loop
- Node.js Frameworks
- Node.js Promise Chaining
- Node.js This Binding
- Node.js Global Objects
- Node.js Automatic Restart Server with Nodemon
- Node.js Child Process
- Node.js Global Installation of Dependencies
- Node.js Session Variable

Node.js Complete References

- Node.js Assert
- Node.js Buffer
- Node.js Console
- Node.js Crypto
- Node.js DNS
- Node.js File System
- Node.js Globals
- Node.js HTTP Module
- Node.js HTTP2
- Node.js OS
- Node.js Path Module
- Node.js Process
- Node.js Query String
- Node.js Stream
- Node.js String Decoder

- Node.js Timers
- Node.js TLS/SSL
- Node.js UDP/DataGram
- Node.js URL
- Node.js Utility
- Node.js V8
- Node.js VM
- Node.js Zlib
- Node.js Questions

Node.js Complete References

Also Check:Recent Articles on Node

Node.js Interview Questions

- Node.js Interview Questions and Answers (2024) ? Beginner Level
- Node.js Interview Questions and Answers (2024) ? Intermediate Level
- Node.js Interview Questions and Answers (2024) ? Advanced Level

Node.js Online Quiz Questions

- Node.js Quiz | Set-1
- Node.js Quiz | Set-2
- Node.js Quiz | Set-3

Node.js Projects

- Library Management System
- User Management System
- Stock Market Portfolio App
- Chat App using socket.io Node.js
- Mail sender using Node.js

- Task Manager using Node.js

Careers with Node.js

Around 6 million websites use Node.js, it is growing day by day. Most of the eCommerce, and IoT companies looking for pro Node.js developer.

Careers with Node.js

- Accenture
- Google
- Cognizant
- HCL Technologies
- Capgemini
- IBM India
- Microsoft India
- Amazon India
- Adobe Inc
- Oracle Inc
- Intel India
- Deloitte India
- Cisco Systems Inc

Node.js Advantages

- Easy Scalability: Node.js compiles and executes JavaScript at lightning speeds, making it highly scalable.
- Real-time Web Apps: Node.js enables real-time communication for chat, gaming, social media updates, and more.
- Microservices: Node.js is lightweight and ideal for microservice architectures.
- JavaScript Everywhere: Learn JavaScript once, and you can use it both for front-end and back-end

development.

- Efficient Data Streaming: Node.js efficiently handles I/O processes like media transcoding during uploads.
- Event-Driven Architecture: Unlike traditional servers, Node.js handles concurrent requests effectively.
- Strong Community Support: Node.js has an independent community backing its development.

Node.js Jobs

If you are curious about what job profiles you will get after learning Node, then here in this section we have listed down some of the job profiles that any Node.js developer can easily get.

Node.js Jobs

- Backend Developer
- Full-Stack Developer
- API Developer
- Microservice Developer
- DevOps Engineer

Node.js vs Other Backend Technologies

Feature | Node.js | Python/Django | PHP

Performance | High for I/O operations | Moderate | Moderate

Scalability | Excellent for microservices | Moderate | Moderate

Learning Curve | Moderate | Easy | Easy

Community Support | Extensive | Extensive | Extensive

Use Cases | Real-time apps, APIs | Web development | CMS, Web development

How to check the version of Node.js?

To check the version of Node.js, open Terminal or CMD and type `node -v` and press Enter.

How to check the version of npm for Node?

Type `npm -v` in CMD or terminal and press Enter to check the version of npm for Node.

Are Node.js developers in demand?

Yes, Node.js developer are in demand because of its rich ecosystem and non blocking architecture

What is the difference between Node.js and React.js?

Well, most of the Node.js is used for backend for any application and React.js is used to create UI of any web application.

What is Node.js is used for?

A misconception among developer is that Node.js is only used for backend but, this is not a whole true. Actually, Node.js is used for both frontend and backend development.

What is Node.js is used for?

Become a Full-Stack Developer and also get 90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

What is Node.js is used for?

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need?Start the challenge right away!

What is Node.js is used for?

S

What is Node.js is used for?

- Web Technologies
- Node.js
- Backend-Development

- Node.js-Basics
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation

- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp

- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux

- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures

- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Java Overview

- Introduction to Java
- The Complete History of Java Programming Language
- How to Install Java on Windows, Linux and macOS?
- Setting up Environment Variables For Java
- How JVM Works - JVM Architecture
- JDK in Java
- Differences between JDK, JRE and JVM

Java Basics

- Java Syntax
- Java Hello World Program
- Java Identifiers
- Java Keywords
- Java Data Types
- Java Variables
- Scope of Variables in Java
- Java Operators
- Java User Input

Java Flow Control

- Java if statement
- Java if-else Statement
- Java if-else-if ladder with Examples
- Java For Loop
- For-each loop in Java
- Java while Loop
- Java Do While Loop

- Java Break Statement
- Java Continue Statement
- Java return Keyword

Java Methods

- Java Methods
- How to Call a Method in Java?
- Static Method vs Instance Method in Java
- Access Modifiers in Java
- Command Line Arguments in Java
- Variable Arguments (Varargs) in Java

Java Arrays

- Arrays in Java
- How to Initialize an Array in Java?
- Java Multi-Dimensional Arrays
- Jagged Array in Java
- Arrays class in Java
- Final Arrays in Java

Java Strings

- Java Strings
- Why Java Strings are Immutable?
- Java String concat() Method with Examples
- String class in Java
- StringBuffer class in Java
- Java StringBuilder Class
- String vs StringBuilder vs StringBuffer in Java

Java OOPs Concepts

- Java OOP(Object Oriented Programming) Concepts
- Classes and Objects in Java
- Java Constructors
- Object Class in Java
- Abstraction in Java
- Encapsulation in Java
- Inheritance in Java
- Polymorphism in Java
- Method Overloading in Java
- Overriding in Java
- Java Packages

Java Interfaces

- Java Interface
- Interfaces and Inheritance in Java
- Java Class vs Interfaces
- Java Functional Interfaces
- Nested Interface in Java
- Marker Interface in Java
- Java Comparator Interface

Java Collections

- Collections in Java
- Collections Class in Java
- Collection Interface in Java
- Java List Interface
- ArrayList in Java

- Vector Class in Java
- LinkedList in Java
- Stack Class in Java
- Set in Java
- Java HashSet
- TreeSet in Java
- Java LinkedHashSet
- Queue Interface In Java
- PriorityQueue in Java
- Deque Interface in Java
- Map Interface in Java
- HashMap in Java
- Java LinkedHashMap
- Hashtable in Java
- Java Dictionary Class
- SortedSet Interface in Java with Examples
- Java Comparator Interface
- Java Comparable Interface
- Java Comparable vs Comparator
- Java Iterator

Java Exception Handling

- Java Exception Handling
- Java Checked vs Unchecked Exceptions
- Java Try Catch Block
- Java final, finally and finalize
- throw and throws in Java

- User-Defined Custom Exception in Java
- Chained Exceptions in Java
- Null Pointer Exception in Java
- Exception Handling with Method Overriding in Java

Java Multithreading

- Java Multithreading Tutorial
- Java Threads
- Java Thread Class
- Java Runnable Interface
- Lifecycle and States of a Thread in Java
- Main thread in Java
- Java Thread Priority in Multithreading
- Java Naming a Thread and Fetching Name of Current Thread
- Java Thread.start() vs Thread.run() Method
- Java Thread.sleep() Method
- Java Daemon Thread
- Java Thread Safety and How to Achieve it?
- Thread Pools in Java

Java File Handling

- File Handling in Java
- Java File Class
- Java Program to Create a New File
- Java Program to Write into a File
- Delete a File Using Java
- Java FileReader Class
- Java FileWriter Class

- Java FilePermission Class
- Java FileDescriptor Class

Java Streams and Lambda Expressions

- Java Lambda Expressions
- Java Method References
- Java 8 Stream Tutorial
- Java 8 Features - Complete Tutorial

Java IO

- Java IO : Input-output in Java with Examples
- Java Reader Class
- Java Writer Class
- Java FileInputStream Class
- FileOutputStream in Java
- Ways to Read Input from Console in Java
- Java BufferedOutputStream Class
- Java BufferedReader vs Scanner Class
- Fast I/O in Java in Competitive Programming

Java Synchronization

- Synchronization in Java
- Importance of Thread Synchronization in Java
- Java Method and Block Synchronization
- Java Atomic vs Volatile vs Synchronized
- Java Lock Framework vs Thread Synchronization
- Deadlock in Java Multithreading
- Deadlock Prevention And Avoidance

- Difference Between Lock and Monitor in Java Concurrency
- ReentrantLock in Java

Java Regex

- Regular Expressions in Java
- Java Pattern pattern() Method
- Java Matcher pattern() Method
- java.lang.Character Class Methods | Set 1
- Java Quantifiers

Java Networking

- Java Networking
- TCP/IP Model
- User Datagram Protocol (UDP)
- Difference Between IPv4 and IPv6
- Difference Between Connection-oriented and Connection-less Services
- Socket Programming in Java
- Java ServerSocket Class
- Java URL Class

JDBC

- JDBC (Java Database Connectivity)
- JDBC Drivers
- Establishing JDBC Connection in Java
- Types of Statements in JDBC

Java Memory Allocation

- Java Memory Management
- How are Java Objects Stored in Memory?

- Stack vs Heap Memory Allocation
- Java Virtual Machine (JVM) Stack Area
- How Many Types of Memory Areas are Allocated by JVM?
- Garbage Collection in Java
- JVM Garbage Collectors
- Stack vs Heap Memory Allocation
- Memory leaks in Java

Java Interview Questions

- Java Interview Questions and Answers
- Java Multiple Choice Questions

Java Practice Problems

- Java Programs - Java Programming Examples
- Java Exercises - Basic to Advanced Java Practice Programs with Solutions
- Java Quiz | Level Up Your Java Skills

Java Projects

- Top 50 Java Project Ideas For Beginners and Advanced [Update 2025]
- Number guessing game in Java
- Mini Banking Application in Java
- Java program to convert Currency using AWT
- Tic-Tac-Toe Game in Java
- Design Snake Game
- Memory Game in Java
- How to Implement a Simple Chat Application Using Sockets in Java?
- Image Processing in Java - Face Detection
- Design Media Sharing Social Networking System

- Java Swing | Create a simple text editor

Java OOP(Object Oriented Programming) Concepts

-
-
-

Java OOP(Object Oriented Programming) Concepts

Object-Oriented Programming or Java OOPs concept refers to programming languages that use objects in programming. They use objects as a primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks you assign.

Java OOP(Object Oriented Programming) Concepts

Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc. in programming. The main aim of OOPs is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Example:

It is a simple example showing a class Numbers containing two variables which can be accessed and updated only by instance of the object created.

Java Class

A Class is a user-defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. Using classes, you can create multiple objects with the same behavior instead of writing their code multiple times. This includes classes for objects occurring more than once in your code. In general, class declarations can include these components in order:

Java Class

- Modifiers: A class can be public or have default access (Refer to this for details).
- Class name: The class name should begin with the initial letter capitalized by convention.
- Body: The class body is surrounded by braces, { }.

Java Object

An Object is a basic unit of Object-Oriented Programming that represents real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. The objects are what perform your code, they are the part of your code visible to the viewer/user. An object mainly consists of:

Java Object

- State: It is represented by the attributes of an object. It also reflects the properties of an object.
- Behavior: It is represented by the methods of an object. It also reflects the response of an object to other objects.
- Identity: It is a unique name given to an object that enables it to interact with other objects.
- Method: A method is a collection of statements that perform some specific task and return the result to the caller. A method can perform some specific task without returning anything. Methods allow us to reuse the code without retyping it, which is why they are considered time savers. In Java, every method must be part of some class, which is different from languages like C, C++, and Python.

Example:

For more information, please refer to the article [?Classes and Object](#).

Method and Method Passing

A method is a collection of statements that perform specific tasks and return a result to the caller. It can be declared with or without arguments, depending on the requirements. A method can take input values, perform operations, and return a result.

4 Pillars of Java OOPs Concepts

1. Abstraction

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or non-essential units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components. Data Abstraction may also be defined as the process of identifying only the required characteristics of an object, ignoring the irrelevant details. The properties and behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the object.

1. Abstraction

Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the car speed or applying brakes will stop the car, but he does not know how on pressing the accelerator, the speed is actually increasing. He does not know about the inner mechanism of the car or the implementation of the accelerators, brakes etc. in the car. This is what abstraction is.

1. Abstraction

Note: In Java, abstraction is achieved by interfaces and abstract classes. We can achieve 100% abstraction using interfaces.

1. Abstraction

Example:

1. Abstraction

To learn more about the Abstraction refer to the [Abstraction in Java](#) article

2. Encapsulation

It is defined as the wrapping up of data under a single unit. It is the mechanism that binds together the code and the data it manipulates. Another way to think about encapsulation is that it is a protective shield that prevents the data from being accessed by the code outside this shield.

2. Encapsulation

- Technically, in encapsulation, the variables or the data in a class is hidden from any other class and can be accessed only through any member function of the class in which they are declared.
- In encapsulation, the data in a class is hidden from other classes, which is similar to what data-hiding does. So, the terms ?encapsulation? and ?data-hiding? are used interchangeably.
- Encapsulation can be achieved by declaring all the variables in a class as private and writing public methods in the class to set and get the values of the variables.

2. Encapsulation

Example:

2. Encapsulation

To learn more about topic refer to [Encapsulation in Java](#) article.

3. Inheritance

Inheritance is an important pillar of OOP (Object Oriented Programming). It is the mechanism in Java by which one class is allowed to inherit the features (fields and methods) of another class. We are achieving inheritance by using `extends` keyword. Inheritance is also known as ?is-a? relationship.

3. Inheritance

Let us discuss some frequently used important terminologies:

3. Inheritance

- Superclass: The class whose features are inherited is known as superclass (also known as base or parent class).
- Subclass: The class that inherits the other class is known as subclass (also known as derived or extended or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- Reusability: Inheritance supports the concept of ?reusability?, i.e. when we want to create a new

class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

3.Inheritance

Example:

3.Inheritance

To learn more about topic refer to [Inheritance in Java](#) article.

4.Polymorphism

It refers to the ability of object-oriented programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of these entities. The ability to appear in many forms is called polymorphism.

4.Polymorphism

Example:

Types of Polymorphism

Polymorphism in Java is mainly of 2 types as mentioned below:

Types of Polymorphism

- Method Overloading
- Method Overriding

Method Overloading and Method Overriding

1. Method Overloading: Also, known as compile-time polymorphism, is the concept of Polymorphism where more than one method share the same name with different signature (Parameters) in a class. The return type of these methods can or cannot be same.

Method Overloading and Method Overriding

2. Method Overriding: Also, known as run-time polymorphism, is the concept of Polymorphism where a method in the child class has the same name, return-type and parameters as in parent class. The child class provides the implementation in the method already written.

Method Overloading and Method Overriding

Below is the implementation of both the concepts:

Method Overloading and Method Overriding

To know more about the topic refer the [Polymorphism in Java](#) article.

Advantage of OOPs over Procedure-Oriented Programming Language

Object-oriented programming (OOP) offers several key advantages over procedural programming:

Advantage of OOPs over Procedure-Oriented Programming Language

- OOP promotes code reusability: By using objects and classes, you can create reusable components, leading to less duplication and more efficient development.
- OOP enhances code organization: It provides a clear and logical structure, making the code easier to understand, maintain, and debug.
- OOP supports the DRY (Don't Repeat Yourself) principle: This principle encourages minimizing code repetition, leading to cleaner, more maintainable code. Common functionalities are placed in a single location and reused, reducing redundancy.
- OOP enables faster development: By reusing existing code and creating modular components, OOP allows for quicker and more efficient application development.

Conclusion

The Object Oriented Programming (OOPs) concept in Java is a powerful way to organize and write code. It uses key ideas like classes, objects, inheritance, polymorphism, encapsulation, and abstraction to create flexible and reusable code.

Conclusion

By using the Java OOPs concept, programmers can build complex applications more efficiently, making the code easier to manage, understand, and modify. Overall, Java's OOPs concepts help in creating robust and scalable software solutions.

What is OOPs concept in Java?

OOPs (Object-Oriented Programming) is a programming paradigm based on the concept of objects, which can contain data in the form of fields (attributes or properties) and code in the form of procedures (methods or functions). In Java, OOPs concepts include encapsulation, inheritance, polymorphism, and abstraction.

Why is OOPs important in Java?

OOPs helps in organizing and structuring code in a more manageable way, making it easier to maintain and scale Java applications. It also promotes code reusability, modularity, and flexibility, leading to efficient and robust software development.

What are the main principles of OOPs in Java?

The main principles of OOPs in Java are encapsulation, inheritance, polymorphism, and abstraction. Encapsulation ensures that the internal state of an object is hidden and can only be accessed through public methods. Inheritance allows one class to inherit properties and behavior from another. Polymorphism enables objects to be treated as instances of their parent class. Abstraction focuses on hiding the implementation details and showing only the necessary information to the outside world.

How is OOPs implemented in Java?

In Java, OOPs is implemented through classes and objects. A class serves as a blueprint for creating objects, which are instances of that class. Each object has its own set of attributes (variables) and methods (functions). By following OOPs concepts like encapsulation, inheritance, polymorphism, and abstraction, Java developers can design well-structured and maintainable code.

What are the advantages of using OOPs in Java?

Some advantages of using OOPs in Java include code reusability, modularity, flexibility, scalability, and easier maintenance. OOPs enables developers to model real-world entities as objects, leading to more intuitive and organized code. It also supports features like inheritance and polymorphism, which enhance the extensibility and readability of Java applications.

Can you provide an example of OOPs concept implementation in Java?

Sure! An example of OOPs concept implementation in Java is creating a `Car` class with attributes like `make`, `model`, and `year`, along with methods like `start()`, `accelerate()`, and `stop()`. By instantiating objects from the `Car` class and calling its methods, we can simulate the behavior of different car instances in a structured and object-oriented manner.

Can you provide an example of OOPs concept implementation in Java?

Start your Java programming journey today with our Java Programming Online Course, designed for both beginners and advanced learners. With self-paced lessons covering everything from basic syntax to advanced concepts, you'll gain the skills needed to excel in the world of programming. Take the 90 Day Challenge! Complete 90% of the course in 90 days, and earn a 90% refund. Track your progress and stay motivated as you master Java. Join now and start your path to becoming a Java expert!

Can you provide an example of OOPs concept implementation in Java?

R

Can you provide an example of OOPs concept implementation in Java?

- Java
- Java-Object Oriented

Can you provide an example of OOPs concept implementation in Java?

- Java

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python

- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP

- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems

- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner

- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming

- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

C++ Overview

- Introduction to C++ Programming Language
- Features of C++
- History of C++
- Interesting Facts about C++
- Setting up C++ Development Environment
- Difference between C and C++

C++ Basics

- Writing First C++ Program - Hello World Example
- C++ Basic Syntax
- C++ Comments
- Tokens in C
- C++ Keywords
- Difference between Keyword and Identifier in C

C++ Variables and Constants

- C++ Variables
- Constants in C
- Scope of Variables in C++
- Storage Classes in C++ with Examples
- Static Keyword in C++

C++ Data Types and Literals

- C++ Data Types
- Literals in C
- Derived Data Types in C++
- User Defined Data Types in C++
- Data Type Ranges and Their Macros in C++
- C++ Type Modifiers
- Type Conversion in C++
- Casting Operators in C++

C++ Operators

- Operators in C++
- C++ Arithmetic Operators
- Unary operators in C

- Bitwise Operators in C
- Assignment Operators in C
- C++ sizeof Operator
- Scope Resolution Operator in C++

C++ Input/Output

- Basic Input / Output in C++
- cin in C++
- cout in C++
- Standard Error Stream Object - cerr in C++
- Manipulators in C++

C++ Control Statements

- Decision Making in C (if , if..else, Nested if, if-else-if)
- C++ if Statement
- C++ if else Statement
- C++ if else if Ladder
- Switch Statement in C++
- Jump statements in C++
- C++ Loops
- for Loop in C++
- Range-Based for Loop in C++
- C++ While Loop
- C++ do while Loop

C++ Functions

- Functions in C++
- return Statement in C++

- Parameter Passing Techniques in C
- Difference Between Call by Value and Call by Reference in C
- Default Arguments in C++
- Inline Functions in C++
- Lambda expression in C++

C++ Pointers and References

- Pointers and References in C++
- C++ Pointers
- Dangling, Void , Null and Wild Pointers in C
- Applications of Pointers in C
- Understanding nullptr in C++
- References in C++
- Can References Refer to Invalid Location in C++?
- Pointers vs References in C++
- Passing By Pointer vs Passing By Reference in C++
- When do we pass arguments by pointer?

C++ Arrays

- Variable Length Arrays (VLAs) in C
- Pointer to an Array | Array Pointer
- How to print size of array parameter in C++?
- Pass Array to Functions in C
- What is Array Decay in C++? How can it be prevented?

C++ Strings

- Strings in C++
- std::string class in C++

- Array of Strings in C++ - 5 Different Ways to Create
- String Concatenation in C++
- Tokenizing a string in C++
- Substring in C++

C++ Structures and Unions

- Structures, Unions and Enumerations in C++
- Structures in C++
- C++ - Pointer to Structure
- Self Referential Structures
- Difference Between C Structures and C++ Structures
- Enumeration in C++
- typedef in C++
- Array of Structures vs Array within a Structure in C

C++ Dynamic Memory Management

- Dynamic Memory Allocation in C using malloc(), calloc(), free() and realloc()
- new and delete Operators in C++ For Dynamic Memory
- new vs malloc() and free() vs delete in C++
- What is Memory Leak? How can we avoid?
- Difference between Static and Dynamic Memory Allocation in C

C++ Object-Oriented Programming

- Object Oriented Programming in C++
- C++ Classes and Objects
- Access Modifiers in C++
- Friend Class and Function in C++
- Constructors in C++

- Default Constructors in C++
- Copy Constructor in C++
- Destructors in C++
- Private Destructor in C++
- When is a Copy Constructor Called in C++?
- Shallow Copy and Deep Copy in C++
- When Should We Write Our Own Copy Constructor in C++?
- Does C++ compiler create default constructor when we write our own?
- C++ Static Data Members
- Static Member Function in C++
- 'this' pointer in C++
- Scope Resolution Operator vs this pointer in C++
- Local Classes in C++
- Nested Classes in C++
- Enum Classes in C++ and Their Advantage over Enum DataType
- Difference Between Structure and Class in C++
- Why C++ is partially Object Oriented Language?

C++ Encapsulation and Abstraction

- Encapsulation in C++
- Abstraction in C++
- Difference between Abstraction and Encapsulation in C++

C++ Polymorphism

- C++ Polymorphism
- Function Overriding in C++
- Virtual Functions and Runtime Polymorphism in C++
- Difference between Inheritance and Polymorphism

C++ Function Overloading

- Function Overloading in C++
- Constructor Overloading in C++
- Functions that cannot be overloaded in C++
- Function overloading and const keyword
- Function Overloading and Return Type in C++
- Function Overloading and float in C++
- Can main() be overloaded in C++?
- Function Overloading vs Function Overriding in C++
- Advantages and Disadvantages of Function Overloading in C++

C++ Operator Overloading

- Operator Overloading in C++
- Types of Operator Overloading in C++
- Functors in C++
- What are the Operators that Can be and Cannot be Overloaded in C++?

C++ Inheritance

- Inheritance in C++
- C++ Inheritance Access
- Multiple Inheritance in C++
- C++ Hierarchical Inheritance
- C++ Multilevel Inheritance
- Constructor in Multiple Inheritance in C++
- Inheritance and Friendship in C++
- Does overloading work with Inheritance?
- Difference between Inheritance and Polymorphism

C++ Virtual Functions

- Virtual Function in C++
- Virtual Functions in Derived Classes in C++
- Default Arguments and Virtual Function in C++
- Can Virtual Functions be Inlined in C++?
- Virtual Destructor
- Advanced C++ | Virtual Constructor
- Advanced C++ | Virtual Copy Constructor
- Pure Virtual Functions and Abstract Classes in C++
- Pure Virtual Destructor in C++
- Can Static Functions Be Virtual in C++?
- RTTI (Run-Time Type Information) in C++
- Can Virtual Functions be Private in C++?

C++ Exception Handling

- Exception Handling in C++
- Exception Handling using classes in C++
- Stack Unwinding in C++
- User-defined Custom Exception with class in C++

C++ Files and Streams

- File Handling through C++ Classes
- I/O Redirection in C++

C++ Templates

- Templates in C++ with Examples
- Using Keyword in C++ STL

C++ Standard Template Library (STL)

- C++ Standard Template Library (STL)
- Containers in C++ STL (Standard Template Library)
- Introduction to Iterators in C++
- Algorithm Library | C++ Magicians STL Algorithm
- Functors in C++

C++ Preprocessors

- C Preprocessors
- C Preprocessor Directives
- #include in C
- C Preprocessors
- C Preprocessor Directives
- Difference between Preprocessor Directives and Function Templates in C++

C++ Namespace

- Namespace in C++ | Set 1 (Introduction)
- namespace in C++ | Set 2 (Extending namespace and Unnamed namespace)
- Namespace in C++ | Set 3 (Accessing, creating header, nesting and aliasing)
- C++ Inline Namespaces and Usage of the "using" Directive Inside Namespaces

Advanced C++

- Multithreading in C++
- Smart Pointers in C++
- auto_ptr vs unique_ptr vs shared_ptr vs weak_ptr in C++
- Type of 'this' Pointer in C++
- "delete this" in C++
- Passing a Function as a Parameter in C++
- Signal Handling in C++

- Generics in C++

C vs C++

- Difference between C and C++
- Difference between C++ and Objective C
- Write a C program that won't compile in C++
- Write a program that produces different results in C and C++
- How does 'void*' differ in C and C++?
- Type Difference of Character Literals in C and C++
- Difference Between C Structures and C++ Structures
- Cin-Cout vs Scanf-Printf

C++ vs Java

- Similarities and Difference between Java and C++
- Comparison of Inheritance in C++ and Java
- Static Keyword in C++
- How Does Default Virtual Behavior Differ in C++ and Java?
- Comparison of Exception Handling in C++ and Java
- Foreach in C++ and Java
- Templates in C++ vs Generics in Java
- Floating Point Operations & Associativity in C, C++ and Java

Competitive Programming in C++

- Competitive Programming - A Complete Guide
- C++ tricks for competitive programming (for C++ 11)
- Writing C/C++ code efficiently in Competitive programming
- Why C++ is best for Competitive Programming?
- Test Case Generation | Set 1 (Random Numbers, Arrays and Matrices)

- Fast I/O for Competitive Programming
- Setting up Sublime Text for C++ Competitive Programming Environment
- How to setup Competitive Programming in Visual Studio Code for C++
- Which C++ libraries are useful for competitive programming?
- Common mistakes to be avoided in Competitive Programming in C++ | Beginners

C++ Interview Questions

- C++ Interview Questions and Answers (2025)
- Top C++ STL Interview Questions and Answers
- 30 OOPs Interview Questions and Answers [2025 Updated]
- Top C++ Exception Handling Interview Questions and Answers
- C++ Programming Examples
- C++ Interview Questions and Answers (2025)

Object Oriented Programming in C++

-
-
-

Object Oriented Programming in C++

Object-oriented programming? As the name suggests uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc. in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Object Oriented Programming in C++

There are some basic concepts that act as the building blocks of OOPs i.e.

Object Oriented Programming in C++

- Class
- Object
- Encapsulation
- Abstraction
- Polymorphism
- Inheritance
- Dynamic Binding
- Message Passing

Characteristics of an Object-Oriented Programming Language

Class

The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object. For Example: Consider the Class of Cars. There may be many cars with different names and brands but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range, etc. So here, the Car is the class, and wheels, speed limits, and mileage are their properties.

Class

- A Class is a user-defined data type that has data members and member functions.
- Data members are the data variables and member functions are the functions used to manipulate these variables together these data members and member functions define the properties and behavior of the objects in a Class.
- In the above example of class Car, the data member will be speed limit, mileage, etc and member functions can apply brakes, increase speed, etc.

Class

We can say that aClass in C++ is a blueprint representing a group of objects which shares some

common properties and behaviors.

Object

An Object is an identifiable entity with some characteristics and behavior. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Object

Objects take up space in memory and have an associated address like a record in pascal or structure or union. When a program is executed the objects interact by sending messages to one another. Each object contains data and code to manipulate the data. Objects can interact without having to know details of each other's data or code, it is sufficient to know the type of message accepted and the type of response returned by the objects.

Object

To know more about C++ Objects and Classes, refer to this article [?C++ Classes and Objects](#)

Encapsulation

In normal terms, Encapsulation is defined as wrapping up data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section, etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of the sales section and the employees that can manipulate them are wrapped under a single name ?sales

section?.

Encapsulation

Encapsulation in C++

Encapsulation

Encapsulation also leads to data abstraction or data hiding. Using encapsulation also hides the data. In the above example, the data of any of the sections like sales, finance, or accounts are hidden from any other section.

Encapsulation

To know more about encapsulation, refer to this article [?Encapsulation in C++](#)

Abstraction

Data abstraction is one of the most essential and important features of object-oriented programming in C++. Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. Consider a real-life example of a man driving a car. The man only knows that pressing the accelerator will increase the speed of the car or applying brakes will stop the car but he does not know how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of an accelerator, brakes, etc. in the car. This is what abstraction is.

Abstraction

- Abstraction using Classes: We can implement Abstraction in C++ using classes. The class helps us to group data members and member functions using available access specifiers. A Class can decide which data member will be visible to the outside world and which is not.
- Abstraction in Header files: One more type of abstraction in C++ can be header files. For example, consider the `pow()` method present in `math.h` header file. Whenever we need to calculate the power

of a number, we simply call the function `pow()` present in the `math.h` header file and pass the numbers as arguments without knowing the underlying algorithm according to which the function is actually calculating the power of numbers.

Abstraction

To know more about C++ abstraction, refer to this article [?Abstraction in C++](#)

Polymorphism

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A person at the same time can have different characteristics. A man at the same time is a father, a husband, and an employee. So the same person possesses different behavior in different situations. This is called polymorphism. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation. C++ supports operator overloading and function overloading.

Polymorphism

- Operator Overloading: The process of making an operator exhibit different behaviors in different instances is known as operator overloading.
- Function Overloading: Function overloading is using a single function name to perform different types of tasks. Polymorphism is extensively used in implementing inheritance.

Polymorphism

Example: Suppose we have to write a function to add some integers, sometimes there are 2 integers, and sometimes there are 3 integers. We can write the Addition Method with the same name having different parameters, the concerned method will be called according to parameters.

Polymorphism

Polymorphism in C++

Polymorphism

To know more about polymorphism, refer to this article [?Polymorphism in C++](#)

Inheritance

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming.

Inheritance

- Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.
- Super Class: The class whose properties are inherited by a sub-class is called Base Class or Superclass.
- Reusability: Inheritance supports the concept of ?reusability?, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

Inheritance

Example: Dog, Cat, Cow can be Derived Class of Animal Base Class.

Inheritance

Inheritance in C++

Inheritance

To know more about Inheritance, refer to this article [?Inheritance in C++](#)

Dynamic Binding

In dynamic binding, the code to be executed in response to the function call is decided at runtime. C++ has virtual functions to support this. Because dynamic binding is flexible, it avoids the drawbacks of static binding, which connected the function call and definition at build time.

Dynamic Binding

Example:

Dynamic Binding

As we can see, the print() function of the parent class is called even from the derived class object.

To resolve this we use virtual functions.

Dynamic Binding

Above Example with virtual Function:

Message Passing

Objects communicate with one another by sending and receiving information. A message for an object is a request for the execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function, and the information to be sent.

Example:

Related Articles:

Example:

- Classes and Objects
- Inheritance
- Access Modifiers
- Abstraction

Advantage of OOPs over Procedure-oriented programming language

Here are key advantages of Object-Oriented Programming (OOP) over Procedure-Oriented Programming (POP):

Advantage of OOPs over Procedure-oriented programming language

- Modularity and Reusability: OOP promotes modularity through classes and objects, allowing for code reusability.
- Data Encapsulation: OOP encapsulates data within objects, enhancing data security and integrity.
- Inheritance: OOP supports inheritance, reducing redundancy by reusing existing code.
- Polymorphism: OOP allows polymorphism, enabling flexible and dynamic code through method overriding.
- Abstraction: OOP enables abstraction, hiding complex details and exposing only essential features

Advantage of OOPs over Procedure-oriented programming language

Advantage of OOPs over Procedure-oriented programming language

Advantage of OOPs over Procedure-oriented programming language

Advantage of OOPs over Procedure-oriented programming language

Master C++ with our comprehensive C++ Programming Course Online, perfect for anyone from beginners to advanced learners. Learn everything from the fundamentals to advanced C++ concepts, and apply your knowledge through practical exercises. Take the Three 90 Challenge! Finish 90% of the course in 90 days, and enjoy a 90% refund. Stay focused, track your progress, and challenge yourself to complete the course! Join today and begin your C++ programming journey!

Advantage of OOPs over Procedure-oriented programming language

V

Advantage of OOPs over Procedure-oriented programming language

- C++
- CPP-Basics
- cpp-class
- cpp-inheritance

Advantage of OOPs over Procedure-oriented programming language

- CPP

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas

- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science

- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

OS Basics

- What is an Operating System?
- Functions of Operating System
- Types of Operating Systems
- Need and Functions of Operating Systems
- Commonly Used Operating System

Structure of Operating System

- Operating System Services
- Introduction of System Call
- System Programs in Operating System
- Operating Systems Structures
- History of Operating System
- Booting and Dual Booting of Operating System

Types of OS

- Batch Processing Operating System
- Multiprogramming in Operating System
- Time Sharing Operating System
- What is a Network Operating System?
- Real Time Operating System (RTOS)

Process Management

- Introduction of Process Management
- Process Table and Process Control Block (PCB)
- Operations on Processes
- Process Schedulers in Operating System
- Inter Process Communication (IPC)

- Context Switching in Operating System
- Preemptive and Non-Preemptive Scheduling

CPU Scheduling in OS

- CPU Scheduling in Operating Systems
- CPU Scheduling Criteria
- Multiple-Processor Scheduling in Operating System
- Thread Scheduling

Threads in OS

- Thread in Operating System
- Threads and its types in Operating System
- Multithreading in Operating System

Process Synchronization

- Introduction of Process Synchronization
- Race Condition Vulnerability
- Critical Section in Synchronization
- Mutual Exclusion in Synchronization

Critical Section Problem Solution

- Peterson's Algorithm in Process Synchronization
- Semaphores in Process Synchronization
- Semaphores and its types
- Producer Consumer Problem using Semaphores | Set 1
- Readers-Writers Problem | Set 1 (Introduction and Readers Preference Solution)
- Dining Philosopher Problem Using Semaphores
- Hardware Synchronization Algorithms : Unlock and Lock, Test and Set, Swap

Deadlocks & Deadlock Handling Methods

- Introduction of Deadlock in Operating System
- Conditions for Deadlock in Operating System
- Banker's Algorithm in Operating System
- Wait For Graph Deadlock Detection in Distributed System
- Handling Deadlocks
- Deadlock Prevention And Avoidance
- Deadlock Detection And Recovery
- Deadlock Ignorance in Operating System
- Recovery from Deadlock in Operating System

Memory Management

- Memory Management in Operating System
- Implementation of Contiguous Memory Management Techniques
- Non-Contiguous Allocation in Operating System
- Compaction in Operating System
- Best-Fit Allocation in Operating System
- Worst-Fit Allocation in Operating Systems
- First-Fit Allocation in Operating Systems
- Fixed (or static) Partitioning in Operating System
- Variable (or Dynamic) Partitioning in Operating System
- Paging in Operating System
- Segmentation in Operating System
- Virtual Memory in Operating System

Page Replacement Algorithms

- Page Replacement Algorithms in Operating Systems
- Program for Page Replacement Algorithms | Set 2 (FIFO)

- Belady's Anomaly in Page Replacement Algorithms
- Optimal Page Replacement Algorithm
- Program for Least Recently Used (LRU) Page Replacement algorithm
- Techniques to handle Thrashing

Storage Management

- Storage Management
- File Systems in Operating System
- File Allocation Methods
- Free Space Management in Operating System
- Disk Scheduling Algorithms
- RAID (Redundant Arrays of Independent Disks)

OS Interview Questions

- Last Minute Notes ? Operating Systems
- Operating System Interview Questions

OS Quiz and GATE PYQ's

- OS Process Management
- OS Memory Management
- OS Input Output Systems
- OS CPU Scheduling
- Deadlock
- 50 Operating System MCQs with Answers

Operating System Tutorial

-
-
-

Operating System Tutorial

An Operating System(OS) is a software that manages and handles hardware and software resources of a computing device.

Operating System Tutorial

- Responsible for managing and controlling all the activities and sharing of computer resources among different running applications.
- A low-level Software that includes all the basic functions like processor management, memory management, file management, etc.
- It mainly acts a government for your system that has different departments to manage different resources.
- Examples are Linux, Unix, Windows 11, MS DOS, Android, macOS and iOS.

Basics :

- Introduction
- Types of OS
- Functions of OS
- 32 vs 64-bit
- System Initialization
- Kernel in OS
- System Call
- Privileged Instructions

Process Scheduling :

- Process Introduction
- Process Creation and Deletion
- States of a Process
- Process Table and Control Block

- Types of Processes in Process Table
- Process Management Introduction
- Process Scheduler
- CPU Scheduling Algorithms
- Preemptive vs Non-Preemptive
- Time spent in Context Switch
- Dispatcher vs scheduler
- Starvation and Aging in Operating Systems

Process Scheduling :

>> Quiz on CPU Scheduling

Process Synchronization

- Inter Process Communication
- Introduction to Process Synchronization
- Critical Section
- Peterson's Algorithm in Process Synchronization
- Lock variable synchronization mechanism
- Semaphores in operating system
- Classical IPC Problems
- Communication using signals in C
- Mutex vs. Semaphore
- Monitors for Process Synchronization
- Dekker's algorithm
- Bakery Algorithm
- Mutex lock for Linux Thread Synchronization
- Priority Inversion

Process Synchronization

>> Quiz on Process Management in OS

Deadlock :

- Deadlock Introduction
- Deadlock Prevention And Avoidance
- Deadlock Detection And Recovery
- Banker's Algorithm
- Deadlock, Starvation, and Livelock
- Resource Allocation Graph (RAG)
- Methods of resource allocation to processes by operating system
- Program for Deadlock free condition in Operating System
- Deadlock detection in Distributed systems
- Techniques used in centralized approach of deadlock detection in distributed systems

Deadlock :

>> Quiz on Deadlock

Multithreading:

- Operating System | Thread
- Threads and its types
- Operating System | User Level thread Vs Kernel Level thread
- Process-based and Thread-based Multitasking
- Multi threading models
- Benefits of Multithreading
- Operating System | Remote Procedure call (RPC)

Memory Management :

- Introduction to memory and memory units

- Memory Management in Operating System
- Buddy System: Memory allocation technique
- Paging
- Segmentation
- Virtual Memory
- Memory Interleaving
- Operating system based Virtualization
- Swap Space
- Page Fault Handling
- Memory Segmentation in 8086 Microprocessor
- Program for Next Fit algorithm in Memory Management
- Overlays in Memory Management
- Belady's Anomaly
- Page Replacement Algorithms
- Second Chance (or Clock) Page Replacement Policy
- Allocating kernel memory (buddy system and slab system)
- Static and Dynamic Libraries | Set 1
- Working with Shared Libraries | Set 1
- Named Pipe or FIFO with example C program
- Tracing memory usage in Linux

Memory Management :

>> Quiz on Memory Management

Disk Management:

- File Systems
- Unix File System
- Implementing Directory Management using Shell Script

- File Directory | Path Name
- Structures of Directory
- File Allocation Methods
- File Access Methods
- Secondary memory
- Secondary memory ? Hard disk drive
- Disk Scheduling Algorithms
- Program for SSTF disk scheduling algorithm
- What exactly Spooling is all about?
- Difference between Spooling and Buffering
- Free space management

Disk Management:

>> Quiz on Input Output Systems

Misc

- Computer Memory
- Introduction to UNIX System
- Microkernel
- Kernel I/O Subsystem (I/O System)
- Important Linux Commands (leave, diff, cal, ncal, locate and ln)
- Process states and Transitions in a UNIX Process
- Introduction to Linux Shell and Shell Scripting
- ?crontab? in Linux with Examples
- indepth and maxdepth in Linux find() command for limiting search to a specific directory.
- Real time systems
- Tasks in Real Time systems
- Monolithic Kernel and key differences from Microkernel

- Get/Set process resource limits in C
- Dual Mode operations in OS

Types of Operating System

- Batch OS (e.g. Transactions Process, Payroll System, etc.)
- Multi-programmed OS(e.g. Windows, UNIX, macOS, etc.)
- Timesharing OS(e.g. Multics, Linux, etc.)
- Real-Time OS(e.g. PSOS, VRTX, etc.)
- Distributed OS(e.g. LOCUS, Solaris, etc.)

Operating System Functions

- Memory and processor Management
- Network Management
- Security Management
- File Management
- Error Detection
- Job Accounting

Why learn Operating Systems?

OS is the most important part of a computer. Through OS users can interact with computer software. It provides an interface between Hardware and CPU. It also provides a platform for the program to run on it and services to users. It performs all the basic tasks required in an application.

Write the top 10 Operating System Examples?

Some most popular OS examples are given below:

Write the top 10 Operating System Examples?

- Windows
- Linux

- MacOS
- ios
- Android
- Ubuntu
- CentOS
- Solaris
- Chrome OS
- Fedora

What are the Advantages of a multiprocessor system?

A multiprocessor system involves the processing of two or more computer programs simultaneously that share the same memory area. It increases reliability.

What is a thread in OS?

A thread is a lightweight process or subprogram which is part of the process or a program. A thread has its own Registers, Stack, State, and Program counter.

What is a thread in OS?

Quick Links :

What is a thread in OS?

- Last Minute Notes (LMNs) | Operating Systems
- Commonly Asked Operating Systems Interview Questions
- ?Quizzes? on Operating Systems !
- ?Practice Problems? on Operating Systems !

What is a thread in OS?

Dreaming of M.Tech in IIT? Get AIR under 100 with our GATE 2026 CSE & DA courses! Get flexible weekday/weekend options, live mentorship, and mock tests. Access exclusive features like All

India Mock Tests, and Doubt Solving?your GATE success starts now!

What is a thread in OS?

- Operating Systems
- GATE CS

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos

- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)

Similar Reads

- [DSA](#)
- [Data Structures](#)
- [Algorithms](#)
- [DSA for Beginners](#)
- [Basic DSA Problems](#)
- [DSA Roadmap](#)
- [DSA Interview Questions](#)
- [Competitive Programming](#)

Similar Reads

- [Data Science & ML](#)
- [Data Science With Python](#)
- [Data Science For Beginner](#)
- [Machine Learning](#)

- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD

- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL

- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management

- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course

- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT

- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

PHP Basics

- How to install PHP in Windows 10?
- How to Install PHP on Linux?
- PHP Syntax
- How to write comments in PHP ?
- PHP Variables
- PHP echo and print
- PHP Data Types
- PHP Strings

PHP Array

- PHP Arrays
- Associative Arrays in PHP
- Multidimensional arrays in PHP
- Sorting Arrays in PHP

PHP Constants

- PHP Constants
- PHP Constant Class
- PHP Defining Constants
- PHP | Magic Constants

PHP Operators

- PHP Operators
- PHP | Bitwise Operators
- PHP | Ternary Operator

PHP Control Statements

- PHP | Decision Making

- PHP switch Statement
- PHP break (Single and Nested Loops)
- PHP continue Statement

PHP Loops

- PHP Loops
- PHP while Loop
- PHP do-while Loop
- PHP for Loop
- PHP foreach Loop

PHP Functions

- PHP | Functions
- PHP Arrow Functions
- Anonymous recursive function in PHP

PHP Advanced

- PHP | Superglobals
- HTTP GET and POST Methods in PHP
- PHP | Regular Expressions
- PHP Form Processing
- PHP Date and Time
- Describe PHP Include and Require
- PHP File Handling
- PHP | Uploading File
- PHP Cookies
- PHP | Sessions
- Implementing callback in PHP

PHP OOPs

- PHP | Classes
- PHP | Constructors and Destructors
- PHP | Access Specifiers
- Multiple Inheritance in PHP
- PHP Constants
- Abstract Classes in PHP
- PHP | Interface
- Static Function in PHP
- PHP | Namespace

MySQL Database

- PHP | MySQL Database Introduction
- PHP | MySQL (Creating Database)
- PHP Database connection
- Connect PHP to MySQL
- PHP | MySQL (Creating Table)
- PHP | Inserting into MySQL database
- PHP | MySQL Select Query
- PHP | MySQL Delete Query
- PHP | MySQL WHERE Clause
- PHP | MySQL UPDATE Query
- PHP | MySQL LIMIT Clause

Complete References

- PHP Array Functions
- PHP String Functions Complete Reference
- PHP Math Functions Complete Reference

- [PHP Filesystem Functions Complete Reference](#)
- [PHP intl Functions Complete Reference](#)
- [PHP IntlChar Functions Complete Reference](#)
- [PHP Image Processing and GD Functions Complete Reference](#)
- [PHP Imagick Functions Complete Reference](#)
- [PHP ImagickDraw Functions Complete Reference](#)
- [PHP Gmagick Functions Complete Reference](#)
- [PHP GMP Functions Complete Reference](#)
- [PHP Ds\Set Functions Complete Reference](#)
- [PHP Ds\Map Functions Complete Reference](#)
- [PHP Ds\Stack Functions Complete Reference](#)
- [PHP Ds\Queue Functions Complete Reference](#)
- [PHP Ds\PriorityQueue Functions Complete Reference](#)
- [PHP Ds\Deque Functions Complete Reference](#)
- [PHP Ds\Sequence Functions Complete Reference](#)
- [PHP SPL Data structures Complete Reference](#)
- [PHP Ds\Vector Functions Complete Reference](#)
- [PHP DOM Functions Complete Reference](#)

PHP Tutorial

-
-
-

PHP Tutorial

PHP (Hypertext Preprocessor) is a versatile and widely used server-side scripting language for creating dynamic and interactive web applications. This PHP tutorial will give you an in-depth understanding of the PHP scripting language. Whether you are a beginner or a professional PHP

developer this free PHP tutorial will provide valuable knowledge about PHP scripting language.

What is PHP?

PHP is a popular scripting language used for creating dynamic web pages and web applications. The term PHP is an acronym of Hypertext Preprocessor. It is an open-source, interpreted, object-oriented server-side scripting language.

What is PHP?

With our PHP tutorial, you'll learn all the important topics, including control statements, functions, arrays, strings, file handling, form handling, regular expressions, date and time manipulation, object-oriented programming in PHP, mathematical operations, working with PHP and MySQL, integrating PHP with Ajax, harnessing the power of PHP with jQuery, and more.

What is PHP?

Table of Content

What is PHP?

- Prerequisites for this PHP Tutorial
- What is PHP?
- Features of PHP
- History of PHP
- PHP Characteristics
- First Hello World Program in PHP
- Why We Learn PHP ?
- Getting Started with PHP Tutorial
- Functions Complete References
- Applications of PHP
- Interview Questions and Answers:
- PHP Online Quiz

- PHP Programming Examples
- PHP Tutorial: Frequently Asked Questions

Prerequisites for this PHP Tutorial

- HTMLandCSS: Understanding of basic HTML and CSS.
- Basic programming concepts

Features of PHP

- Open-Source and Free: PHP is firstly open source which means anyone can use PHP code without any licensing. Along with this one can run PHP on any operating system like Windows, macOS, Linux, Unix and more.
- PHP Server-Side Scripting: PHP code executes on the server before sending HTML content to the user's browser, allowing for the dynamic generation of web pages and handling user interactions.
- Interpreted language: PHP code is interpreted line by line, eliminating the need for compilation and simplifying development and testing processes.
- Database connectivity: PHP integrates seamlessly with various databases like MySQL, PostgreSQL, and Oracle, facilitating data storage and retrieval for web applications.
- Object-oriented programming (OOP): PHP supports OOP concepts like classes, objects, inheritance, and polymorphism, enabling better code organization and modularity.
- Built-in functions: PHP comes with a rich set of built-in functions for various tasks such as string manipulation, date and time handling, file handling, and more, reducing the need for external libraries.
- Session management: PHP allows for user session management, enabling personalized experiences and storing user data across multiple page visits.
- Security features: While security considerations are essential for any development language, PHP offers several built-in security features and best practices to help mitigate vulnerabilities.

History of PHP

PHP is developed by Rasmus Lerdorf in 1994, the very first version of PHP that was simply designed to set the Common Gateway Interface (CGI) binaries, which are written in C programming language. The latest version of PHP is PHP version 8, which is released on November 24, 2022. It can be easily embedded with HTML files. HTML codes can also be written in a PHP file. The PHP codes are executed on the server side, whereas HTML codes are directly executed on the browser.

PHP Characteristics

- Simple
- Efficient
- Secure
- Flexible

First Hello World Program in PHP

Simple program to print "Hello world!" message on the screen.

First Hello World Program in PHP

Output:

Why We Learn PHP ?

PHP is one of the widely used open-source general-purpose scripting languages for backend Development. Apart from this, let's see why we should learn it.

Why We Learn PHP ?

- Easy to Learn: It is easier to learn for anyone who has come across any programming language for the first time.
- Free of Cost: Since it is an open-source language, therefore developers are allowed to use its components and all methods for free.
- Flexible: Since it is a dynamically typed language, therefore there are no hard rules on how to build features using it.

- Supports nearly all databases:It supports all the widely used databases, including MySQL, ODBC, SQLite etc.
- Secured:It has multiple security levels and provides us with a secure platform for developing websites as it has multiple security levels.
- Huge Community Support:It is loved and used by a huge number of developers. The developers share their knowledge with other people in the community who want to know about it.

Why We Learn PHP ?

These applications demonstrate the versatility and widespread use of PHP in web development, powering a vast array of websites and applications.

Why We Learn PHP ?

Also Check:Recent Articles on PHP

Getting Started with PHP Tutorial

Explore the power of the web with this, free PHP tutorial! We'll guide you through the essential building blocks, transforming you from a beginner to a confident PHP developer. Get ready to craft dynamic web applications and robust server-side functionality with the versatile PHP language!

Functions Complete References

Functions Complete References

- Array Functions
- String Functions
- Math Functions
- GMP Functions
- Calendar Functions
- IntlChar Functions
- Imagick Functions

- Gmagick Functions
- ImagickDraw Functions
- Image Processing and GD Functions

Functions Complete References

- SPL Data Structures
- DS\Sequence Functions
- DS\Vector Functions
- Ds\Deque Functions
- Ds\Map Functions
- Ds\Set Functions
- Ds\Stack Functions
- Ds\Queue Functions
- Ds\PriorityQueue Functions
- Filesystem Functions

Applications of PHP

- Server-side web development:It is a development where the program runs on a server dealing with the generation of content of web pages.
- Content management systems (CMS):It is a framework already designed by other programmers and coders on which you can either contribute your knowledge and skills or just use those coders? skills to design your own website or blog
- E-commerce websites:E-commerce, or electronic commerce, refers to the buying and selling of goods and services over the Internet.
- Database-driven applications:It is a software application that relies on a database to store, manage, and retrieve data. It utilizes a database management system (DBMS) to organize and manipulate data, enabling efficient data storage, retrieval, and management.
- Web APIs:It is an API as the name suggests, it can be accessed over the web using the HTTP

protocol. It is a framework that helps you to create and develop HTTP-based RESTFUL services.

Interview Questions and Answers:

- PHP Interview Questions and Answers Set-1
- PHP Interview Questions and Answers Set-2

PHP Online Quiz

- PHP Quiz | Set-1
- PHP Quiz | Set-2
- PHP Quiz | Set-3

PHP Programming Examples

Here in the section, we have listed some PHP programming examples that will help you to level up your PHP programming skills.

PHP Programming Examples

- Programming Examples

What is PHP full form?

The full form of PHP is Hypertext Preprocessor. It was abbreviated previously as Personal Home Page.

Is PHP easy to learn?

Yes, to get a pro in PHP it will take 6 to 12 months so it more towards the easy programming language to learn.

Can you learn PHP on your own

Yes, you one can easily learn PHP by its own, just get some good resource like GeeksforGeeks and YouTube

Why is PHP used for?

PHP is used for a wide range of web development tasks, but here are some of its most common applications:

What is a PHP developer's salary?

The average salary of the PHP developer lies between 3.5 Lakh to 11 Lakh.

What is a PHP developer's salary?

Become a Full-Stack Developer and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

What is a PHP developer's salary?

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

What is a PHP developer's salary?

- PHP
- Web Technologies

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution

- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures

- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs

- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker

- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy

- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced

- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course

- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE

- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Basics

- PostgreSQL Tutorial
- What is PostgreSQL - Introduction
- Install PostgreSQL on Windows
- Install PostgreSQL on Mac

Database Operations

- PostgreSQL - Create Database
- PostgreSQL - Loading a Database
- PostgreSQL ALTER DATABASE
- PostgreSQL - Rename Database
- PostgreSQL - Show Databases

Data Types

- PostgreSQL - Data Types
- PostgreSQL - Boolean Data Type
- PostgreSQL - CHAR Data Type
- PostgreSQL - VARCHAR Data Type
- PostgreSQL - NUMERIC Data Type
- PostgreSQL - Date Data Type
- PostgreSQL - TIME Data Type
- PostgreSQL - JSON Data Type
- PostgreSQL - CREATE DOMAIN

Querying Tables

- PostgreSQL - SELECT
- PostgreSQL - ORDER BY clause
- PostgreSQL - WHERE clause

- PostgreSQL FETCH Clause
- PostgreSQL - IN operator
- PostgreSQL - HAVING clause
- PostgreSQL - GROUP BY clause
- PostgreSQL - LIKE operator
- PostgreSQL - BETWEEN Operator

Table Operations

- PostgreSQL - CREATE TABLE
- PostgreSQL - SELECT INTO
- PostgreSQL - CREATE SEQUENCE
- PostgreSQL - ALTER TABLE
- PostgreSQL - ADD COLUMN
- PostgreSQL - DROP COLUMN
- PostgreSQL - Rename Table
- PostgreSQL - DROP TABLE
- PostgreSQL - TRUNCATE TABLE
- PostgreSQL - Copy a Table
- PostgreSQL - Comparing Tables
- PostgreSQL - Show Tables

Modifying Data

- PostgreSQL - INSERT
- PostgreSQL - Insert Multiple Values in Various Rows
- PostgreSQL UPDATE Statement
- PostgreSQL - DELETE
- PostgreSQL - Upsert

Conditionals

- PostgreSQL - CASE
- PostgreSQL COALESCE
- PostgreSQL - NULLIF() Function
- PostgreSQL - CAST

Control Flow

- PostgreSQL - IF Statement
- PostgreSQL - CASE Statement
- PostgreSQL - Loop Statement
- PostgreSQL - While Loops
- PostgreSQL - Exit Statement
- PostgreSQL - Continue

Transactions & Constraints

- PostgreSQL - Transactions
- PostgreSQL - COMMIT
- PostgreSQL - Primary Key
- PostgreSQL - Foreign Key
- PostgreSQL - CHECK Constraint
- PostgreSQL - UNIQUE Constraint
- PostgreSQL - NOT NULL Constraint

JOINS & Schemas

- PostgreSQL - Joins
- PostgreSQL - LEFT JOIN
- PostgreSQL - INNER JOIN
- PostgreSQL - FULL OUTER JOIN

- PostgreSQL - SELF JOIN
- PostgreSQL - Schema
- PostgreSQL - CREATE SCHEMA
- PostgreSQL - DROP SCHEMA
- PostgreSQL - ALTER SCHEMA

Roles & Permissions

- PostgreSQL - CREATE ROLE
- PostgreSQL - ALTER ROLE
- PostgreSQL - DROP ROLE
- PostgreSQL - GRANT
- PostgreSQL - REVOKE
- PostgreSQL - Role Membership

Operators

- PostgreSQL - UNION operator
- PostgreSQL - INTERSECT Operator
- PostgreSQL - EXCEPT Operator
- PostgreSQL - ANY Operator
- PostgreSQL - ALL Operator
- PostgreSQL - EXISTS Operator

PostgreSQL Functions

- PostgreSQL - CREATE FUNCTION Statement
- PostgreSQL - Function Overloading
- PostgreSQL - DROP FUNCTION
- PostgreSQL - MAX() Function
- PostgreSQL MIN() Function

- PostgreSQL - SUM() Function
- PostgreSQL - COUNT() Function
- PostgreSQL - EXTRACT Function
- PostgreSQL - REPLACE() Function

Errors & Exception

- PostgreSQL - Errors and Messages
- Exception Handling in PL/SQL
- PostgreSQL - Assert

Advance Concepts

- PostgreSQL - Dollar-Quoted String Constants
- PostgreSQL - Block Structure
- PostgreSQL - Variables
- PostgreSQL - Introduction to Stored Procedures
- PostgreSQL - Trigger
- PostgreSQL - CREATE TRIGGER
- PostgreSQL - DROP TRIGGER
- PostgreSQL - Disabling a Trigger
- PostgreSQL - Enabling a Trigger
- PostgreSQL - CREATE INDEX
- PostgreSQL - List Indexes
- PostgreSQL - UNIQUE Index

PostgreSQL Tutorial

-
-
-

PostgreSQL Tutorial

In this PostgreSQL tutorial you'll learn the basic data types(Boolean, char, text, time, int etc.), Querying and Filtering techniques like select, where, in, order by, etc. managing and modifying the tables in PostgreSQL. We'll cover all the basic to advance concepts of PostgreSQL in this tutorial. So if you are beginner who start to learn RDBMS or an expert who want to upscale you PostgreSQL skills, then keep reading this free PostgreSQL tutorial.

PostgreSQL Tutorial

PostgreSQL is one of the most advanced Relational database management systems (RDBMS). It is open-source software, which means the source code is available under the PostgreSQL license. Anyone with the right skills is free to use, modify, and distribute PostgreSQL in any form. It supports both relational as well as Non-Relational JSON Queries.

PostgreSQL Tutorial

What is PostgreSQL?

PostgreSQL is an advanced and open-source relational database management system and is used as a database for many web applications, mobile and analytics applications. It supports both SQL (relational) and JSON (non-relational) querying and It is a stable database supported by more than 20 years of development by the open-source community.

What is the Use of PostgreSQL?

As you know that PostgreSQL is a powerful, open-source relational database management system. It's used which is mostly used for storing, organizing, and retrieving data. Other than this things it is also widely used for:

What is the Use of PostgreSQL?

- Data Storage: It stores data in an organized way using tables, making it easy to manage large amounts of information.

- **Complex Queries:**PostgreSQL supports complex queries, allowing you to fetch specific data using SQL (Structured Query Language).
- **ACID Compliance:**It ensures reliable transactions with ACID (Atomicity, Consistency, Isolation, Durability) compliance, meaning your data remains accurate and secure.
- **Scalability:**Suitable for both small and large applications, PostgreSQL can handle a large number of users and large datasets.
- **Extensibility:**You can add custom functions, data types, and extensions, making PostgreSQL highly flexible and adaptable to various needs.
- **Support for Advanced Data Types:**It supports advanced data types likeJSON,XML, and arrays, allowing for diverse data storage and manipulation.

What is the Use of PostgreSQL?

Table of Content

What is the Use of PostgreSQL?

- What is PostgreSQL?
- PostgreSQL Basics
- Data Types
- Querying & Filtering Data
- Managing Tables
- Modifying Data
- Conditionals
- Control Flow
- Transactions & Constraints
- Working with JOINS & Schemas
- Roles & Permissions
- Working with Sets
- Subquery & CTEs

- User-Defined Functions
- Important In-Built Functions
- PostgreSQL PL/pgSQL
- Variables & Constants
- Stored Procedures
- Working with Triggers
- Working with Views & Indexes
- Errors & Exception Handling
- Features of PostgreSQL
- Advantages of PostgreSQL

PostgreSQL Tutorial For Beginners

In this beginner section, you'll learn the basics, from installing PostgreSQL to creating your first database and running simple queries. We'll guide you step-by-step through setting up tables, inserting data, and retrieving information using SQL commands.

PostgreSQL Basics

Explore all the basics topics that every one know while learning the PostgreSQL, Here in this section you will learn introduction of PostgreSQL, how to install PostgreSQL on various OS and other PostgreSQL basics.

PostgreSQL Basics

- PostgreSQL ? Introduction
- Install PostgreSQL on Windows
- Install PostgreSQL on Mac
- PostgreSQL ? Loading a Database
- PostgreSQL ? Create Database
- PostgreSQL ? ALTER DATABASE

- PostgreSQL ? DROP DATABASE
- PostgreSQL ? Rename Database
- PostgreSQL ? Copy Database
- PostgreSQL ? Size of a Database
- PostgreSQL ? Show Databases

Data Types

- PostgreSQL ? Data Types
- PostgreSQL ? Boolean Data Type
- PostgreSQL ? CHAR Data Type
- PostgreSQL ? VARCHAR Data Type
- PostgreSQL ? TEXT Data Type
- PostgreSQL ? NUMERIC Data Type
- PostgreSQL ? SMALLINT Integer Data Type
- PostgreSQL ? INTEGER Data Type
- PostgreSQL ? BIGINT Integer Data Type
- PostgreSQL ? Date Data Type
- PostgreSQL ? Timestamp Data Type.
- PostgreSQL ? UUID Data Type
- PostgreSQL ? TIME Data Type
- PostgreSQL ? Interval Data Type
- PostgreSQL ? User-defined Data Type
- PostgreSQL ? Array Data Type
- PostgreSQL ? hstore Data Type
- PostgreSQL ? JSON Data Type

Querying & Filtering Data

- PostgreSQL ? SELECT

- PostgreSQL ? SELECT DISTINCT clause
- PostgreSQL ? ORDER BY clause
- PostgreSQL ? WHERE clause
- PostgreSQL ? LIMIT clause
- PostgreSQL ? FETCH clause
- PostgreSQL ? IN operator
- PostgreSQL ? IS NULL operator
- PostgreSQL ? LIKE operator
- PostgreSQL ? NOT LIKE operator
- PostgreSQL ? BETWEEN operator
- PostgreSQL ? HAVING clause
- PostgreSQL ? GROUP BY clause

Managing Tables

- PostgreSQL ? CREATE TABLE
- PostgreSQL ? SELECT INTO
- PostgreSQL ? Create Auto-increment Column using SERIAL
- PostgreSQL ? CREATE SEQUENCE
- PostgreSQL ? Identity Column
- PostgreSQL ? ALTER TABLE
- PostgreSQL ? Rename Table
- PostgreSQL ? ADD COLUMN
- PostgreSQL ? DROP COLUMN
- PostgreSQL ? Change Column Type
- PostgreSQL ? RENAME COLUMN
- PostgreSQL ? DROP TABLE
- PostgreSQL ? TRUNCATE TABLE

- PostgreSQL ? Temporary Table
- PostgreSQL ? Temporary table name
- PostgreSQL ? Removing Temporary Table
- PostgreSQL ? Copy a Table
- PostgreSQL ? Comparing Tables
- PostgreSQL ? Describe Table
- PostgreSQL ? Show Tables
- PostgreSQL ? Import CSV File Into Table

Modifying Data

- PostgreSQL ? INSERT
- PostgreSQL ? Insert multiple rows
- PostgreSQL ? UPDATE
- PostgreSQL ? DELETE
- PostgreSQL ? Upsert

Conditionals

- PostgreSQL ? CASE
- PostgreSQL ? COALESCE
- PostgreSQL ? NULLIF() Function
- PostgreSQL ? CAST

Control Flow

- PostgreSQL ? IF Statement
- PostgreSQL ? CASE Statement
- PostgreSQL ? Loop Statement
- PostgreSQL ? While Loops
- PostgreSQL ? For Loops

- PostgreSQL ? Exit
- PostgreSQL ? Continue

Transactions & Constraints

- PostgreSQL ? Transactions
- PostgreSQL ? COMMIT
- PostgreSQL ? BEGIN
- PostgreSQL ? ROLLBACK
- PostgreSQL ? Primary Key
- PostgreSQL ? Foreign Key
- PostgreSQL ? CHECK Constraint
- PostgreSQL ? UNIQUE Constraint
- PostgreSQL ? NOT NULL Constraint

Working with JOINS & Schemas

- PostgreSQL ? Joins
- PostgreSQL ? LEFT JOIN
- PostgreSQL ? INNER JOIN
- PostgreSQL ? FULL OUTER JOIN
- PostgreSQL ? SELF JOIN
- PostgreSQL ? Schema
- PostgreSQL ? CREATE SCHEMA
- PostgreSQL ? DROP SCHEMA
- PostgreSQL ? ALTER SCHEMA

Roles & Permissions

- PostgreSQL ? CREATE ROLE
- PostgreSQL ? ALTER ROLE

- PostgreSQL ? DROP ROLE
- PostgreSQL ? GRANT
- PostgreSQL ? REVOKE
- PostgreSQL ? Role Membership

Working with Sets

- PostgreSQL ? UNION operator
- PostgreSQL ? INTERSECT Operator
- PostgreSQL ? EXCEPT Operator
- PostgreSQL ? GROUPING SETS
- PostgreSQL ? CUBE
- PostgreSQL ? ROLLUP

Subquery & CTEs

- PostgreSQL ? ANY Operator
- PostgreSQL ? ALL Operator
- PostgreSQL ? EXISTS Operator
- PostgreSQL ? CTE
- PostgreSQL ? Deleting Duplicate Rows using Subquery

User-Defined Functions

- PostgreSQL ? CREATE FUNCTION Statement
- PostgreSQL ? Function parameter modes
- PostgreSQL ? Function Overloading
- PostgreSQL ? Function that returns table
- PostgreSQL ? Drop Function

Important In-Built Functions

- PostgreSQL ? AVG() Function

- PostgreSQL ? COUNT() Function
- PostgreSQL ? MAX() Function
- PostgreSQL ? MIN() Function
- PostgreSQL ? SUM() Function
- PostgreSQL ? FIRST_VALUE Function
- PostgreSQL ? LAST_VALUE Function
- PostgreSQL ? NTH_VALUE Function
- PostgreSQL ? ROW_NUMBER Function
- PostgreSQL ? CURRENT_DATE Function
- PostgreSQL ? CURRENT_TIME Function
- PostgreSQL ? EXTRACT Function
- PostgreSQL- CONCAT Function
- PostgreSQL ? FORMAT Function
- PostgreSQL ? UPPER function
- PostgreSQL- LOWER function
- PostgreSQL ? REGEXP_MATCHES Function
- PostgreSQL ? REGEXP_REPLACE Function
- PostgreSQL ? REPLACE Function

Important In-Built Functions

Visit [PostgreSQL In-Built functions](#) for more.

Advanced PostgreSQL Tutorial

In this section, you'll learn about advanced features of PostgreSQL that can help you manage your databases more efficiently. We'll cover topics like performance tuning, indexing strategies, and using advanced SQL functions. You'll also explore how to handle large datasets, optimize queries, and set up replication for high availability.

PostgreSQL PL/pgSQL

- PostgreSQL ? Dollar-Quoted String Constants
- PostgreSQL ? Block Structure

Variables & Constants

- PostgreSQL ? Variables
- PostgreSQL ? Select Into
- PostgreSQL ? Row type variables
- PostgreSQL ? Record type variable
- PostgreSQL ? Constants

Stored Procedures

- PostgreSQL ? Introduction to Stored Procedures
- PostgreSQL ? CREATE PROCEDURE
- PostgreSQL ? Drop Procedure

Working with Triggers

- PostgreSQL ? Trigger
- PostgreSQL ? CREATE TRIGGER
- PostgreSQL ? DROP TRIGGER
- PostgreSQL ? ALTER TRIGGER
- PostgreSQL ? Disabling a Trigger
- PostgreSQL ? Enabling a Trigger

Working with Views & Indexes

- PostgreSQL ? CREATE INDEX
- PostgreSQL ? DROP INDEX
- PostgreSQL ? List Indexes
- PostgreSQL ? UNIQUE Index

- PostgreSQL ? Partial Index
- PostgreSQL ? Index On Expression
- PostgreSQL ? REINDEX
- PostgreSQL ? Multicolumn Indexes

Errors & Exception Handling

- PostgreSQL ? Errors and Messages
- PostgreSQL ? Exception Handling
- PostgreSQL ? Assert

Difference Between PostgreSQL and MySQL

Feature | PostgreSQL | MySQL

Type | Object-relational database | Relational database

ACID Compliance | Fully ACID compliant | Fully ACID compliant

Complex Queries | Excellent support for complex queries | Good support for complex queries

Data Types | Wide range of advanced data types | Basic range of data types

JSON Support | Strong support for JSON | Good support for JSON

Performance | Great for complex, read-heavy operations | Fast for read-heavy and write-heavy loads

Extensibility | Highly extensible with custom functions | Limited extensibility

Replication | Supports various replication methods | Supports master-slave replication

Community and Support | Strong community, active development | Strong community, widely used

Usage | Preferred for complex applications | Preferred for web applications

License | Open-source (PostgreSQL License) | Open-source (GPL License)

Features of PostgreSQL

PostgreSQL runs on all operating systems, Like Linux, UNIX, MAC OS and Windows and It supports text, images, sounds, and video, and includes programming interfaces for C / C++, Java, Perl,

Python, Ruby, and Open Database Connectivity (ODBC).

Features of PostgreSQL

PostgreSQL supports a big part of the SQL standard and provides many features mentioned below:

Features of PostgreSQL

- Complex SQL queries
- SQL Sub-selects
- Foreign keys
- Trigger
- Views
- Transactions
- Multiversion concurrency control (MVCC)
- Streaming Replication (as of 9.0)
- Hot Standby (as of 9.0)
- Asynchronous replication
- Tablespaces

Advantages of PostgreSQL

- PostgreSQL has the feature of write-ahead logging.
- Many replication methods are supported.
- It has ability to make large-scale web applications because it is robust and powerful.
- It is easy to learn.
- According to the organization we can edit and modify it easily because PostgreSQL is available for free to its open source license.

Conclusion

In this PostgreSQL tutorial, You will learn all the essentials of working with PostgreSQL Like installation, connecting to the database, creating and managing databases, SQL basics, querying

data, advanced queries and joins, indexing and optimization, transactions and concurrency control, security and user management, backup and restore, PostGIS for geospatial data, and PostgreSQL extensions. PostgreSQL provides number of features Which makes it a versatile and reliable choice for building robust database-driven applications.

1. How to create new Database in PostgreSQL?

There are two methods of creating a new database:

1. How to create new Database in PostgreSQL?

- CREATE DATABASE(SQL command)
- createdb(command-line executable)

2. What are the different Operators in PostgreSQL?

There are 4 different type of operators in PostgreSQL are as follow-

2. What are the different Operators in PostgreSQL?

- Arithmetic operators
- Logic operators
- Comparison operators
- Bitwise operators

3. What are the disadvantages with PostgreSQL?

Performance wise PostgreSQL is slower than MySQL and Open source applications are less than MySQL.

4. How to delete a PostgreSQL database?

- DROP DATABASE(SQL command)
- dropdb(command-line)

4. How to delete a PostgreSQL database?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

4. How to delete a PostgreSQL database?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

4. How to delete a PostgreSQL database?

- Databases
- PostgreSQL
- postgresQL

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon

- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- [More Tutorials](#)
- [Software Development](#)
- [Software Testing](#)
- [Product Management](#)
- [Project Management](#)
- [Linux](#)
- [Excel](#)
- [All Cheat Sheets](#)
- [Recent Articles](#)

Similar Reads

- [Free Online Tools](#)
- [Typing Test](#)
- [Image Editor](#)
- [Code Formatters](#)
- [Code Converters](#)
- [Currency Converter](#)
- [Random Number Generator](#)
- [Random Password Generator](#)

Similar Reads

- [Write & Earn](#)
- [Write an Article](#)
- [Improve an Article](#)
- [Pick Topics to Write](#)
- [Share your Experiences](#)
- [Internships](#)

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Mathematical/Analytical Puzzle

- Puzzle 1 | (How to Measure 45 minutes using two identical wires?)
- Puzzle - Find the ages of daughters
- Puzzle - Calculate total distance travelled by bee
- Monty Hall problem
- Puzzle 16 | (100 Doors)
- Puzzle - Torch and Bridge
- Puzzle - 2 Eggs and 100 Floors
- Puzzle 12 | (Maximize probability of White Ball)
- Puzzle 27 | (Hourglasses Puzzle)
- Puzzle 17 | (Ratio of Boys and Girls in a Country where people want only boys)
- Puzzle 29 | (Car Wheel Puzzle)
- Puzzle 22 | (Maximum Chocolates)
- Puzzle 28 | (Newspaper Puzzle)
- Puzzle 33 | (Rs 500 Note Puzzle)
- Puzzle 39 | (100 coins puzzle)
- Puzzle 44 | Girl or Boy
- Puzzle 26 | (Know Average Salary without Disclosing Individual Salaries)
- Puzzle 37 | (Maximum run in cricket)
- Puzzle 32 | (Completion of Task)
- Puzzle 40 | (Find missing Row in Excel)
- Four People on a Rickety Bridge
- Puzzle | Man fell in well Puzzle
- Puzzle | 50 red marbles and 50 blue marbles
- Puzzle | Bag of Coins
- Puzzle | Find the last ball to remain after the entire process
- Puzzle | 10 identical bottles of pills

- Hungry Worm and Old Tree Riddle | Puzzle
- Puzzle 85 | Chain Link Puzzle
- Interview Puzzle | The shopkeeper and the lady who made a purchase of Rs 200 with fake note
- Puzzle | Snail and Wall
- Puzzle 45 | Gem in Pockets
- Pizza Puzzle
- Puzzle 49 | King and his Elephants
- Puzzle | Find the box which contains Box of 11 gm Cigarettes
- Puzzle | How much he had initially?
- Puzzle | Minimum planes to go around the world
- Puzzle | Six colored cube

Logical Puzzles

- Puzzle 9 | (Find the fastest 3 horses)
- Puzzle 5 | (Finding the Injection for Anesthesia)
- Puzzle 4 | (Pay an employee using a gold rod of 7 units ?)
- Puzzle 7 | (3 Bulbs and 3 Switches)
- Puzzle 15 | (Camel and Banana Puzzle)
- Puzzle 8 | (Find the Jar with contaminated pills)
- Puzzle 10 | (A Man with Medical Condition and 2 Pills)
- Puzzle 13 | (100 Prisoners with Red/Black Hats)
- Puzzle 24 | (10 Coins Puzzle)
- Puzzle 14 | (Strategy for a 2 Player Coin Game)
- Puzzle 19 | (Poison and Rat)
- Puzzle 20 | (5 Pirates and 100 Gold Coins)
- Puzzle 31 | (Minimum cut Puzzle)
- Puzzle 34 | (Prisoner and Policeman Puzzle)

- Puzzle 36 | (Matchstick Puzzle)
- Missionaries and Cannibals
- Puzzle 41 | (Guess Color of Hat)
- Puzzle 43 | Muddy Heads
- Puzzle | Heaven and Hell
- Puzzle | Mislabeled Jars
- Puzzle | 8 balls problem
- Puzzle 51| Cheryl's Birthday Puzzle and Solution
- Puzzle | Measure 4L using given 3 buckets
- Puzzle | 3 Priests and 3 devils Puzzle
- Puzzle | Farmer, Goat, Wolf and Cabbage
- Puzzle | Water Jug Problem
- Puzzle | Blind man and Pills
- Puzzle | The Burning Candles
- Birthday Puzzle
- Puzzle | Rat and Poisonous Milk Bottles
- Measuring 6L water from 4L and 9L buckets
- Puzzle | Six Houses P, Q, R, S, T, and U
- Melting Candles | Puzzle
- Puzzle 47 | Red Hat vs Blue Hat
- Puzzle | Find the Culprit
- Puzzle 54 | Fill the Jug
- Puzzle 46 | Rich or Poor
- Puzzle | Light all the bulbs
- Puzzle | Distribute the Water
- Puzzle 50 | The Boat Wreckage
- Puzzle | Weight of Heavy Ball

- Puzzle | Guess the bit string

Arrangement Puzzles

- Puzzle 11 | (1000 Coins and 10 Bags)
- Puzzle 23 | (Days of month using 2 dice)
- Puzzle 38 | (Tic Tac Toe Puzzle)
- Puzzle 30 | (Last Palindrome Date Before 10/02/2001)
- Puzzle 42 | (Placing the numbers)
- Puzzle | 10 Balls in 5 Lines
- Puzzle | (Round table coin game)
- Puzzle | Place numbers 1 to 9 in a Circle such that sum of every triplet in straight line is 15

Shape based Puzzles

- Puzzle 21 | (3 Ants and Triangle)
- Puzzle | 3 cuts to cut round cake into 8 equal pieces
- Puzzle 25 | (Chessboard and dominos)
- Puzzle | Three Squares
- Puzzle | Maximum pieces that can be cut from a Circle using 6 straight lines
- Puzzle | Splitting a Cake with a Missing Piece in two equal portion
- Puzzle 48 | Rectangular Cardboard
- Puzzle | Dividing a Square into N smaller squares

Rubik's Cube

- Algorithm to solve Rubik's Cube
- Rubik's Cube Facts and Questions

Brain Teasers

- Brain Teasers
- 20 Challenging Brain Teasers with Answers

Crossword Puzzle

- Crossword Puzzle Of The Week #1 (for DSA)
- Crossword Puzzle Of The Week #2 (for Computer Science and Applications)
- Crossword Puzzle Of The Week #3 (for Database and Queries)
- Crossword Puzzle Of The Week #4 (for Object Oriented Programming)
- Crossword Puzzle Of The Week #5 (for Operating System)

Puzzles

-
-
-

Puzzles

Puzzles are commonly asked in exams and interviews to test logical and analytical thinking. Here is a list of most asked Puzzles divided into four categories as per examination pattern.

1. Analytical / Mathematical Puzzles

Puzzle Title | Asked in Company

Find ages of daughters | Google,Microsoft

Calculate total distance traveled by bee | Yahoo

6x6 Grid - Ways to Reach Bottom Right | Amazon,Zoho

Monty Hall problem | VMWare

Torch and Bridge | Google,Microsoft

2 Eggs and 100 Floors | VMWare

Maximize the probability of White Ball | Amazon

Poison and Rat | Amazon

Hourglasses Puzzle | Bank of America,Yahoo

The ratio of Boys and Girls in a Country where people want only boys | Google,Goldman Sachs

Car Wheel Puzzle | MakeMytrip

Maximum Chocolates | Infosys,MakeMytrip

Puzzle | Splitting a Cake with a Missing Piece in two equal portion | Alcatel-Lucent,Cognizant

Rs 500 Note Puzzle | CAT,UPSC

Girl or Boy | Amazon

Know Average Salary without Disclosing Individual Salaries | Infosys, Bloomberg.

Maximum run in cricket | FAANG

Completion of Task | Reflexis Systems

Find missing Row in Excel | philips

Four People on a Rickety Bridge | Jumbotail,SAP

Man fell in well Puzzle | American Express

50 red marbles and 50 blue marbles | Google,Microsoft, JP morgan chase

Puzzle | Form Three Equilateral Triangles | Google

10 identical bottles of pills | ZS Associate

Puzzle | Maximum pieces that can be cut from a Circle using 6 straight lines | TCS

Chain Link Puzzle | cognizant

The shopkeeper and the lady who made a purchase of Rs 200 with fake note | Persistent Systems

Egg Dropping Puzzle with 2 Eggs and K Floors | Google,Microsoft

Minimum number of Apples to be collected from trees to guarantee M red apples | Leetcode

Snail and Wall | TCS

1000 light bulbs switched on/off by 1000 people passing by | UK university interview

Puzzle | Four Alternating Knights | Amazon,Google

TCS DIGITAL PUZZLE | Lateral Thinking 2 | TCS

Puzzle | 100 Cows And Milk | ZS

Puzzle | One Mile on the Globe | Microsoft

TCS DIGITAL PUZZLE | Lateral Thinking | TCS

Puzzle | The Counters and Board | JP-morgan

Camel and Banana Puzzle | Amazon,Yahoo

Puzzle | (Six Matches , Right Foot Forward) | TCS

How much he had initially? | IAS interview question

Puzzle | 3 cuts to cut round cake into 8 equal pieces | Adobe,Citius Tech,Cognizant,blackrock

Two Creepers Climbing a Tree | Adobe,Google,Microsoft

1. Analytical / Mathematical Puzzles

Puzzle Title

1. Analytical / Mathematical Puzzles

Asked in Company

1. Analytical / Mathematical Puzzles

Read More:Articles on Analytical/mathematical puzzles

2. Logical Puzzles

Puzzle Title | Asked in Company

Pay an employee using a gold rod of 7 units ? | FAANG, Ola cabs

Find the fastest 3 horses | Accolite,Goldman Sachs,MakeMyTrip

Finding the injection for Anesthesia | Google,Yahoo

3 Bulbs and 3 Switches | MakeMyTrip,Qualcomm

Camel and Banana Puzzle | Amazon,Yahoo

Find the Jar with contaminated pills | MakeMyTrip

100 Prisoners with Red/Black Hats | Google,Microsoft

10 Coins Puzzle | Google,Yahoo

Strategy for a 2-Player Coin Game | TCS

5 Pirates and 100 Gold Coins | Microsoft

Minimum cut Puzzle | Amazon

Prisoner and Policeman Puzzle | Microsoft

Puzzle - Cheating Husband | Microsoft,Google

Puzzle ? Blind Games | Bloomberg L.P.

Puzzle ? Chameleons go on a date | Amazon

Heaven and Hell | Amazon,Infosys

Mislabeled Jars | Google,Microsoft

8 balls problem | Microsoft, Simence

Cheryl's Birthday Puzzle and Solution | Facebook, Whatsapp, Singapore math Olympic

Puzzle ? The Lion and the Unicorn | The Access Group (UK),TCS

Farmer, Goat, Wolf, and Cabbage | Infosys

Water Jug Problem | Wells Fargo

Blind man and Pills | Mentor Graphics

The Burning Candles | Wipro

Puzzle | The Burning Candles | Wipro,IBM,TCS

Rat and Poisonous Milk Bottles | Google

Measuring 6L water from 4L and 9L buckets | Microsoft

Six Houses P, Q, R, S, T, and U | CAT Quiz

Melting Candles | Faang

Red Hat vs Blue Hat | Microsoft

Puzzle | Joint family of seven persons (L, M, N, O, P, Q, and R) | TCS

Puzzle | The Circle of Lights | Microsoft, Bloomberg

Puzzle | 9 Students and Red Black Hats | Google

Light all the bulbs | Microsoft, Bloomberg

Distribute the Water | Microsoft

Puzzle | Can 2 persons be with same number of hairs on their heads? | oppo, inflame

Weight of Heavy Ball | IBM

2. Logical Puzzles

Puzzle Title

2. Logical Puzzles

Asked in Company

2. Logical Puzzles

Read More:Articles on Logical Puzzles

3. Arrangement Puzzles

Puzzle Title | Asked in Company

10 Coins Puzzle | Google,Yahoo

Days of the month using 2 dice | Microsoft

Tic Tac Toe Puzzle | Amazon

Matchstick Puzzle | Kirloskar Brothers, SLB

Last Palindrome Date Before 10/02/2001 | Amazon

10 identical bottles of pills | ZS Associate

10 Balls in 5 Lines | Publicis Sapient

Round table coin game | ElectrifiAi

Puzzle | The Circle of Lights | Microsoft, Bloomberg

3. Arrangement Puzzles

Puzzle Title

3. Arrangement Puzzles

Asked in Company

3. Arrangement Puzzles

Read More:Articles on Arrangement Puzzles

3. Shape based Puzzles

Puzzle Title | Asked in Company

3 cuts to cut the round cake into 8 equal pieces | Adobe,Citius Tech,Cognizant

Chessboard and dominos | Google

Puzzle 36 | (Matchstick Puzzle) | SLB

Maximum pieces that can be cut from a Circle using 6 straight lines | TCS

Splitting a Cake with a Missing Piece in two equal portion | SAP

3 Ants and Triangle | Intuit, ZS Associate,EXL

3. Shape based Puzzles

Puzzle Title

3. Shape based Puzzles

Asked in Company

3. Shape based Puzzles

Read More:Articles on Shape based puzzles

4. Mechanical Puzzles

- Algorithm to solve Rubik's Cube
- Crossword Puzzle Of The Week #1 (for DSA)
- Crossword Puzzle Of The Week #2 (for Computer Science and Applications)
- Crossword Puzzle Of The Week #3 (for Database and Queries)
- Crossword Puzzle Of The Week #4 (for Object Oriented Programming)

Quick Links:

- Logic Building Coding Problems
- 'Practice Problems' on Puzzles
- Recent Puzzles!

Quick Links:

Get 90% Course fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today. The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Quick Links:

- Puzzles
- Tutorials

Quick Links:

- Puzzles

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon

- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Python Basics

- Python Syntax
- Python Comments
- Python Variables
- Python Keywords

Data Types

- Python String
- Python Lists
- Python Tuples
- Python Sets
- Dictionaries in Python
- Python Arrays

Python Tutorial | Learn Python Programming Language

-
-
-

Python Tutorial | Learn Python Programming Language

Python Tutorial- Python is one of the most popular programming languages today, known for its simplicity, extensive features and library support. Its clean and straightforward syntax makes it beginner-friendly, while its powerful libraries and frameworks makes it perfect for developers.

Python Tutorial | Learn Python Programming Language

- A high-level, interpreted language with easy-to-read syntax.
- Used in various fields likeweb development, data science, artificial intelligenceandautomation.

First Python Program

Here is a simple Python code, printing a string. We recommend you to edit the code and try to print your own name.

1. Python Fundamentals

In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords and operators. These are essential building blocks to get started with Python coding.

1. Python Fundamentals

Before starting to learn python we need to install python on our system.

1. Python Fundamentals

- Introduction
- Input and Output
- Variables
- Keywords
- Operators
- Quizzes :Fundamentals,Input/Output

2. Python Data Types

Python offers versatile collections of data types, including lists, string, tuples, sets, dictionaries and arrays. In this section, we will learn about each data types in detail.

2. Python Data Types

- Numbers
- Boolean
- Strings
- Type Casting
- List

- Tuples
- Dictionary
- Sets
- Arrays
- Quiz:Data Types

3. Python Conditional Statements and Loops

Python Conditional statements are used to make decisions based on certain conditions. They allow the program to execute specific blocks of code when a condition evaluates to True. Python Loops are used to execute a block of code repeatedly, either for a fixed number of times or until a condition is met. In this section of Python Tutorial, we'll explore Python's conditional logic and Python loops. Here, we'll explore Python loop constructs, including for and while loops, along with essential loop control statements like break, continue and pass. Additionally, we'll uncover list and dictionary comprehensions for concise iterations and data manipulations .

3. Python Conditional Statements and Loops

- Conditional Statements
- For Loop
- While Loop
- Loop control statements (break, continue, pass)
- List Comprehension
- Quizzes:Control Flow,Loops

4. Python Functions

Python Functions are the backbone of organized and efficient code in Python. Here, in this section of Python 3 tutorial we'll explore their syntax, parameter handling, return values and variable scope. From basic concepts to advanced techniques like closures and decorators. Along the way, we'll also introduce versatile functions like range(), map, filter and lambda functions.

4. Python Functions

- def keyword
- Use of pass Statement in Function
- Return statement
- Global and Local Variables
- Recursion in Python
- *args and **kwargs in Function
- ?Self? as Default Argument
- First Class Function
- Lambda Function
- Map,ReduceandFilter Function
- Inner Function
- Decorators
- Quiz:Functions

5. Python OOPs Concepts

In this section ofPython OOPs, we'll explore the core principles of object-oriented programming (OOP) in Python. From encapsulation to inheritance, polymorphism, abstract classes and iterators, we'll cover the essential concepts that empower you to build modular, reusable and scalable code.

5. Python OOPs Concepts

- Classes and Objects
- Polymorphism
- Inheritance
- Abstract
- Encapsulation
- Iterators
- Quiz:OOPs

6. Python Exception Handling

In this section of Python Tutorial, we'll explore Python Exception Handling that how Python deals with unexpected errors, enabling you to write robust and fault-tolerant code. We'll cover file handling, including reading from and writing to files.

6. Python Exception Handling

- Exception handling
- Try and Except
- Built-in Exception
- User defined Exception
- Quiz:Exception Handling

7. File Handling

In this section, we will cover file handling, including reading from and writing to files.

7. File Handling

- File Handling
- Different File Modes
- Read Files
- Write/Create Files
- OS Module
- pathlib Module
- Directory Management
- Quiz:File Handling

8. Python Collections

Here in this section of Python 3 tutorial, you will explore key data structures provided by Python's collections module.

8. Python Collections

- Counters
- Heapq
- Deque
- OrderedDict
- Defaultdict

9. Python Database Handling

In this section you will learn how to access and work with MySQL and MongoDB databases

9. Python Database Handling

- Python MongoDB Tutorial
- Python MySQL Tutorial

10. Python Packages or Libraries

The biggest strength of Python is a huge collection of Python Packages standard libraries which can be used for the following:

10. Python Packages or Libraries

- Built-in Modules in Python
- Python DSA Libraries
- Machine Learning
- Python GUI Libraries
- Web Scraping Packages
- Game Development Packages
- Web Frameworks like, Django, Flask
- Image processing (like OpenCV, Pillow)

Python Practice

- Exercise with Practice Questions
- List Exercise
- String Exercise
- Dictionary Exercise
- Set Exercise
- Tuple Exercise

Python Quizzes

Python quiz page covers topics including variables, data types and how to manage output effectively. You'll explore operators and control flow to structure our code, along with loops (for and while) for repetitive tasks. Additionally, you'll gain knowledge with Python data structures such as lists, tuples, dictionaries and sets.

Python Quizzes

- Quizzes

Python Quizzes

This Python tutorial is updated based on latest Python 3.13.1 version.

Python Prerequisite

This Python tutorial is designed for beginners, but it helps if you already have some basic knowledge of programming concepts like variables, commands and syntax. A little experience with any programming environment will make the learning easier, but don't worry if you're completely new ? we'll guide you through it step by step.

Features of Python

Python stands out because of its simplicity and versatility, making it a top choice for both beginners and professionals. Here are some key features or characteristics:

Features of Python

- **Easy to Read and Write:** Python's syntax is clean and simple, making the code easy to understand and write. It is suitable for beginners.
- **Interpreted Language:** Python executes code line by line, which helps in easy debugging and testing during development.
- **Object-Oriented and Functional:** Python supports both object-oriented and functional programming, giving developers flexibility in how they structure their code.
- **Dynamically Typed:** You don't need to specify data types when declaring variables; Python figures it out automatically.
- **Extensive Libraries:** Python has a rich collection of libraries for tasks like web development, data analysis, machine learning and more.
- **Cross-Platform:** Python can run on different operating systems like Windows, macOS and Linux without modification.
- **Community Support:** Python has a large, active community that continuously contributes resources, libraries and tools, making it easier to find help or solutions.

Applications of Python

- **Web Development:** Frameworks like Django and Flask can be used to create dynamic websites and web applications quickly and efficiently.
- **Data Science & Analysis:** Python is most preferred language for data analysis, visualization and handling large datasets. Because of extensive libraries like Pandas, NumPy and Matplotlib.
- **Machine Learning & AI:** Python is popular in AI and machine learning because of its powerful libraries like TensorFlow, Keras and Scikit-learn.
- **Scripting & Automation:** Python's simplicity makes it ideal for writing scripts that automate tasks in different systems, from server management to file handling. Python is commonly used to automate repetitive tasks, making processes faster and more efficient.
- **Web Scraping:** Libraries like BeautifulSoup and Scrapy.
- **Desktop App Development:** Python can be used to build desktop applications using frameworks like

Tkinter and PyQt. Python is also used for game development, with libraries like Pygame to create simple games.

Python vs. Other Programming Languages

Below is the comparison of Python with C, C++ andJava:

Python vs. Other Programming Languages

Feature	Python	C	C++	Java
Type	Interpreted	Compiled	Compiled	Compiled and Interpreted
Paradigm	Multi-paradigm (object-oriented, procedural, functional)	Procedural, structured	Multi-paradigm (procedural, object-oriented, generic)	Object-oriented, structured
Memory Management	Automatic	Manual	Manual	Automatic
Syntax	Simple	Complex	Complex	Complex
Use Cases	Web development, data analysis, machine learning	System programming, embedded systems, game development	System programming, game development, high-performance applications	Large-scale applications, enterprise software
Notable Frameworks/Libraries	Django, Flask	Standard Library	Standard Library, Boost	Spring, Hibernate
Community Support	Strong	Strong	Strong	Strong
Job Market	Abundant	Abundant	Abundant	Abundant

List of Companies Using Python

These are some Popular companies that use Python in their workflow:

List of Companies Using Python

Company	Description
Google	Uses Python for various applications, including their search engine and machine learning projects.
Instagram	The backend of Instagram is built using Python, enabling it to handle millions of users

efficiently.

Spotify | Python is used for data analysis and backend services, helping improve user recommendations.

Dropbox | Python powers the desktop client of Dropbox, making it easy to sync files across devices.

Netflix | Python helps Netflix with data analysis and managing its content recommendation algorithms.

Reddit | One of the largest online communities, Reddit, uses Python for its core functionalities.

Uber | Uber uses Python for various features, including dynamic pricing and data analysis.

Pinterest | Python plays a key role in the backend of Pinterest, helping scale and manage user content.

Career & Jobs in Python

Python offer diverse opportunities across industries, here we have listed all the best career opportunity that anyone can pursue after learning Python.

Career & Jobs in Python

Career | Average Salary (INR) Per Annum | Average Salary (USD) Per Annum

Python Developer | ₹500,000 ? ₹1,200,000 | \$60,000 ? \$110,000

Data Scientist | ₹600,000 ? ₹1,500,000 | \$70,000 ? \$130,000

Machine Learning Engineer | ₹700,000 ? ₹1,800,000 | \$75,000 ? \$140,000

Full Stack Developer | ₹600,000 ? ₹1,300,000 | \$65,000 ? \$120,000

DevOps Engineer | ₹800,000 ? ₹2,000,000 | \$80,000 ? \$140,000

Automation Engineer | ₹500,000 ? ₹1,200,000 | \$55,000 ? \$100,000

Data Analyst | ₹400,000 ? ₹900,000 | \$50,000 ? \$90,000

Software Engineer | ₹500,000 ? ₹1,500,000 | \$65,000 ? \$120,000

Backend Developer | ₹600,000 ? ₹1,300,000 | \$70,000 ? \$125,000

AI Engineer | ₹900,000 ? ₹2,500,000 | \$90,000 ? \$160,000

Python Latest & Upcoming Features

Python 3.13 is the most recent stable release of the programming language, featuring a blend of updates to its syntax, implementation and standard library. Key enhancements include the introduction of a revamped interactive interpreter, experimental capabilities for operating in a free-threaded mode (as outlined in PEP 703) and the addition of a Just-In-Time (JIT) compiler (detailed in PEP 744).

Python Latest & Upcoming Features

The upcoming version of Python is Python 3.14 and it is expected to include the following notable updates: UTF-8 Mode Default (PEP 686), Shorthand Syntax for Keyword Arguments, Enhanced f-strings, JSON-based Simple API (PEP 691).

Python Latest & Upcoming Features

--> In this tutorial, we've provided the latest Python 3.13 version compiler where you can edit and compile your written code directly with just one click of the RUN Button. So test yourself with Python first exercises.

What is Python?

Python is a high-level, general-purpose and very popular programming language which was created by Guido van Rossum and released in 1991. Python programming language (latest Python 3) is being used in web development and Machine Learning, Mathematics and System Scripting. Python language is being used by almost all tech-giant companies like ? Google, Amazon, Facebook, Instagram, Dropbox, Uber? etc.

Is it easy to learn Python?

Yes, Python is considered one of the easiest programming languages to learn, especially for beginners.

Is Python enough to get a job?

Yes, knowing Python can be enough to get a job, especially in certain fields. Python is a versatile programming language widely used across various industries and it is highly valued for its simplicity, readability and powerful libraries. Doing specialization is always preferred, you can go for high demand domains like Data Science, AI or Web development.

What is the salary of Python Developer?

Due to high demand of domains like Data science and web dev, Python developer can get good package for India or countries like United States. The estimated salary for a Python Developer is ₹500,000 ? ₹1,200,000 per year. For countries like USA, estimated pay for a Python Developer is \$60,000 ? \$110,000 per Year.

What are job opportunity with Python?

After completing Python you can opt for various careers like:

What are job opportunity with Python?

- Python Developer
- Web Developer
- Data Scientist
- Data Analyst
- Machine Learning Engineer
- AI Researcher
- Automation Engineer
- Software Engineer
- DevOps Engineer

What are the key advantages of learning Python?

There are multiple key advantages of learning Python programming language and they are:

What are the key advantages of learning Python?

- Easy to Learn: Simple syntax, perfect for beginners.
- Versatile: Used in web development, data science, AI and more.
- In-Demand: High job market demand with strong salaries.
- Rich Libraries: Extensive libraries for diverse tasks.
- Cross-Platform: Runs on all major operating systems.

What are the key advantages of learning Python?

Learn Python from basics to advanced with our Python Full Course Online. Gain hands-on experience with practice problems and build your skills step by step. Join the Three 90 Challenge: finish 90% of the course in 90 days and get a 90% refund. Start now and master Python!

What are the key advantages of learning Python?

- Python
- Python Programs
- Tutorials
- python

What are the key advantages of learning Python?

- python
- python

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers

- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

React Fundamentals

- React JSX
- ReactJS Babel Introduction
- ReactJS Virtual DOM
- React JS ReactDOM
- React Lists
- React Forms
- ReactJS Keys
- ReactJS Refs
- ReactJS Rendering Elements
- React Conditional Rendering

React Components

- Code Splitting in React
- React Components
- ReactJS | Components - Set 2
- ReactJS Pure Components
- ReactJS Functional Components
- React Lifecycle
- Differences between Functional Components and Class Components
- ReactJS Container and Presentational Pattern in Components

React Props & States

- ReactJS Methods as Props
- ReactJS PropTypes
- ReactJS Props - Set 1
- ReactJS Props - Set 2
- ReactJS Unidirectional Data Flow

- ReactJS State
- ReactJS State vs Props
- Implementing State in React Components

React Hooks

- React Hooks
- React useState Hook
- ReactJS useEffect Hook
- Context in React
- React Router
- React JS Types of Routers
- ReactJS Fragments

React Apps

- Create ToDo App using ReactJS
- Create a Quiz App using ReactJS
- Create a Coin Flipping App using ReactJS
- How to create a Color-Box App using ReactJS?
- Dice Rolling App using ReactJS
- Guess the number with React

React Connection & Deployment

- How to Deploy Your React Websites on GitHub?
- How to Deploy React project on Firebase?
- How to deploy React app to Heroku?
- How to deploy React app to Surge ?
- How to deploy simple frontend server-less (static) React applications on Netlify

React Exercises

- React Exercises, Practice Questions and Solutions

React Questions

- How to Connect Django with Reactjs ?
- 7 React Best Practices Every Web Developer Should Follow
- 8 Ways to Style React Components
- How to add Stateful component without constructor class in React?
- How to display a PDF as an image in React app using URL?
- React JS Toast Notification
- What is the use of data-reactid attribute in HTML ?
- How to zoom-in and zoom-out image using ReactJS?
- How to avoid binding by using arrow functions in callbacks in ReactJS?
- How to bind 'this' keyword to resolve classical error message 'state of undefined' in React?
- How To Get The Height And Width Of An Image Using ReactJS?
- How To Handle Multiple Input Field In React Form With A Single Function?
- How To Handle States Of Mutable Data Types?
- How to change state continuously after a certain amount of time in React?
- How to change the state of react component on click?

React Tutorial

-
-
-

React Tutorial

React is a JavaScript Library known for front-end development (or user interface). It is popular due to its component-based architecture, Single Page Applications (SPAs) and Virtual DOM for building web applications that are fast, efficient, and scalable.

React Tutorial

- Applications are built using reusable components that enable the reload of only the changed part of the UI. Hence we get a better user experience with React.
- Allows you to describe how the UI should look based on the state of your application.
- Uses a virtual DOM to optimize updates, making applications faster and more efficient.
- Enforces a one-way data flow, making the application predictable and easier to debug.
- It uses JSX which combines HTML-like syntax with JavaScript's functionality, making it easy to shift to React from JavaScript.

React Tutorial

To Start with React you need to install it in your project. Follow these articles to install depending on your system

React Tutorial

- How to Install ReactJS on Windows
- How to Install ReactJS on MacOS
- How to Install ReactJS on Linux

React Tutorial

Let us now take a look at our first code example.

React Tutorial

It will print 'Welcome To React Tutorial' in the Browser.

React Tutorial

- JSX looks like HTML but is JavaScript. `<div>` and `<h1>` are React elements created under the hood.
- Every tag in JSX (like `<h1>` or `<div>`) must be properly closed, including self-closing ones like ``.

- All returned elements must be inside one parent (e.g., the <div> here).
- App is a function that returns the UI. It's exported so it can be used elsewhere.
- import React from 'react'; is required to use JSX, as JSX is transformed into JavaScript using React.

Why Learn React JS?

React, the popular JavaScript library, offers several exciting reasons for developers to learn it.

Why Learn React JS?

- **Flexibility:** React allows you to build high-quality user interfaces across different platforms, thanks to its flexible library approach.
- **Great Developer Experience:** React makes it easier to write and understand code, offering a smooth development process.
- **Strong Support from Facebook:** Regular updates, bug fixes, and resources from Facebook ensure React stays up-to-date and reliable.
- **Vast Community Support:** React benefits from a large community, providing enough resources, tutorials, and troubleshooting help.
- **Excellent Performance:** React's efficient Virtual DOM ensures fast rendering and high performance for web applications.
- **Easy Testing:** React's structure and tools make testing simpler, improving code reliability and maintainability.

Why Learn React JS?

One of the most effective ways to solidify your understanding is through practical application. That's where structured learning comes into play. For those looking to deepen their React skills and progress from beginner to advanced levels, consider enrolling in the course on React JS: Beginner to Advanced offered by GeeksforGeeks. This comprehensive course provides hands-on projects and real-world applications, making it an excellent resource for anyone serious about mastering React.

Why Learn React JS?

React Tutorial Prerequisites:HTML,CSSandJavaScript

React Basic Concepts

Introduction to the Fundamentals of React form importing, exporting modules, creating components, to managing data flow and rendering UI across the application.

React Basic Concepts

- Introduction
- Import and Export
- JSX Introduction
- Components
- Conditional Rendering
- PropTypes
- Prop Drilling
- React Lists
- Context API
- React Redux

React Hooks

Different hooks to access and mange state and lifecycle in functional components.

React Hooks

- Hooks Introduction
- useState Hook
- useEffect Hook
- useRef Hook
- useMemo Hook
- useContext Hook

React DOM Events

Handle DOM events in react like mouse, keyboard and form events.

React DOM Events

- React Events Introduction
- onclickcapture Event
- onMouseDown Event
- onDoubleClick Event
- onSubmit Event
- onScroll Event
- onBlur Event

Lifecycle of Components

The lifecycle of react components from creation, mounting and unmounting to manage the different phases.

Lifecycle of Components

- Introduction to lifecycle of components
- constructor
- render
- componentDidMount
- componentWillUnmount
- componentDidCatch
- componentDidUpdate
- shouldComponentUpdate

Routing in React

Manage navigation and routing within the single page application.

Routing in React

- React JS Router
- What is react-router-dom?
- React JS Types of Routers
- React-Router Hooks
- Learn Navigation and Routing in React Apps
- Link and NavLink components in React-Router-Dom

Important React Packages

Additional libraries for styling and state management to enhance your react projects.

Important React Packages

- Redux
- Material UI
- react-bootstrap
- Tailwind
- Framer Motion

React Interview Questions

React Interview questions categorised for different levels including most asked questions.

React Interview Questions

- Beginner Level Interview Questions (2024)
- Intermediate Level Interview Questions (2024)
- Advanced Level Interview Questions (2024)
- 7 Most Asked ReactJS Interview Questions

React Online Quizzes

Test and Improve your knowledge with the well organised React Quizzes

React Online Quizzes

- Set-1
- Set-2
- Set-3
- Set-4
- Set-5
- Set-6

React Online Practice Exercise

Embark on your React learning journey with our online practice portal. Start by selecting quizzes tailored to your skill level. Engage in hands-on coding exercises, receive real-time feedback, and monitor your progress. With our user-friendly platform, mastering React becomes an enjoyable and personalized experience.

React Online Practice Exercise

Elevate your coding expertise by going through our carefully curated Free Online React Exercises.

React Complete References

Refer to the following articles to have a quick glance at all the important key concepts which are helpful while developing web applications using React

React Complete References

- Basic Concepts Reference
- Components Complete Reference
- Props Reference
- Events Reference
- Hooks Reference
- Advanced Guide Reference

React Complete References

If you want to have a simple, quick reference to commonly used React methods you can refer to [ReactJS Cheat Sheet](#) article

Advantages of React

- Composable: React allows building reusable components, improving code organization and maintainability.
- Declarative: React's declarative approach makes it easy to design interactive UIs by changing component states, with React handling the DOM updates.
- SEO-Friendly: React's Single Page Application (SPA) setup can be optimized for SEO by rendering and indexing content properly.
- Large Community: React has a strong community and is widely used by major companies like Meta, Netflix, making it easier to find resources and support.
- Easy to Learn: React uses JSX, which resembles HTML, making it simple for developers with basic knowledge of HTML, CSS, and JavaScript.
- Easy Debugging: React's unidirectional data flow simplifies debugging by ensuring that data moves in a clear, predictable direction within components.

Advantages of React

If you want to learn more refer to this article [React JS Advantages](#)

What is React used for?

The primary goal of React is to create a User Interface, Single-Page Application, Progressive Web Application and more.

Which language is used in React?

React used JavaScript programming language.

Is React a front-end language?

React globally used for Front-end JS library and it is popular with both software and project speoncer.

Which is in demand ReactJS or AngularJS in 2024?

React is far more popular than AngularJS because ReactJS comes with more libraries, resources and bigger community support.

Which is better NodeJS or ReactJS or AngularJS

Choosing the framework always depends very much on your requirements. All of these three frameworks are quite popular, and users choose them based on what they want to do. But if we do so more specifically, then ReatJS is better.

Is this ReactJS tutorial for beginners or for advanced

As we said at the beginning of this article, this course is for beginners as well as for advanced.

Is this ReactJS tutorial for beginners or for advanced

Become a Full-Stack Developer and also get 90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

Is this ReactJS tutorial for beginners or for advanced

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need?Start the challenge right away!

Is this ReactJS tutorial for beginners or for advanced

S

Is this ReactJS tutorial for beginners or for advanced

- ReactJS
- Web Technologies

- Web-Tech Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy

- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Redis Basics

Redis Data Structures

Redis Commands

Redis Advanced

Introduction to Redis

-
-
-

Introduction to Redis

Redis is an in-memory data structure that is used for faster access to data. It is used to store data that needs to be accessed frequently and fast. It is not used for storing large amounts of data. If you want to store and retrieve large amounts of data you need to use a traditional database such as MongoDB or MYSQL. Redis provides a variety of data structures such as sets, strings, hashes, and lists.

Introduction to Redis

- The Redis server is a program that runs and stores data in memory.
- You can just connect to that server and can use it to store and retrieve data faster.
- For that reason, Redis is not used for persistent storing of data as complete data will be lost if the system crashes.
- Redis is scalable as you can run multiple instances of the server.
- It is often used as a cache that stores data temporarily and provides faster access to frequently used data.

Introduction to Redis

Introduction to Redis

Introduction to Redis

Important Topics for Redis Server

Introduction to Redis

- When to use Redis Server?
- Advantages of Redis Server
- Disadvantages of Redis Server
- How to Start Redis Server?
- Redis Basics
- Redis Data Structures
- Redis Commands
- Redis Advanced

When to use Redis Server?

Consider you have a MySQL database and you are constantly querying the database which reads the data from the secondary storage, computes the result, and returns the result.

When to use Redis Server?

If the data in the database is not changing much you can just store the results of the query in redis-server and then instead of querying the database which is going to take 100-1000 milliseconds, you can just check whether the result of the query is already available in redis or not and return it result which is going to be much faster as it is already available in the memory.

When to use Redis Server?

Note: In a messaging app, Redis can be used to store the last five messages that the user has sent and received using the built-list data structure provided in Redis.

Advantages of Redis Server

- High Performance: Redis excels in terms of performance due to its in-memory nature. It can deliver extremely fast read and write operations, making it suitable for scenarios where low-latency is critical.
- Redis excels in terms of performance due to its in-memory nature. It can deliver extremely fast read and write operations, making it suitable for scenarios where low-latency is critical.
- Simple and Easy-to-Use API: Redis has a straightforward API that consists of simple and intuitive commands, making it easy for developers to use and integrate into their applications.
- Redis has a straightforward API that consists of simple and intuitive commands, making it easy for developers to use and integrate into their applications.
- Data Structures: Redis supports a variety of data structures, including strings, lists, sets, hashes, and more. This versatility allows developers to model their data more effectively, choosing the right data structure for the task at hand.
- Redis supports a variety of data structures, including strings, lists, sets, hashes, and more. This versatility allows developers to model their data more effectively, choosing the right data structure for the task at hand.
- Atomic Operations: Redis supports atomic operations on these data structures, making it a great fit for scenarios that require consistency and reliability in multi-step operations.
- Redis supports atomic operations on these data structures, making it a great fit for scenarios that require consistency and reliability in multi-step operations.
- Persistence Options: While Redis is an in-memory database, it provides persistence options such as snapshots and append-only files. This allows users to configure the level of durability needed for their specific use case.
- While Redis is an in-memory database, it provides persistence options such as snapshots and append-only files. This allows users to configure the level of durability needed for their specific use case.
- Replication and High Availability: Redis supports master-slave replication, enabling the creation of replicas of the master server. This provides high availability and fault tolerance in case the master

node fails.

- Redis supports master-slave replication, enabling the creation of replicas of the master server. This provides high availability and fault tolerance in case the master node fails.

Advantages of Redis Server

- Redis excels in terms of performance due to its in-memory nature. It can deliver extremely fast read and write operations, making it suitable for scenarios where low-latency is critical.

Advantages of Redis Server

- Redis has a straightforward API that consists of simple and intuitive commands, making it easy for developers to use and integrate into their applications.

Advantages of Redis Server

- Redis supports a variety of data structures, including strings, lists, sets, hashes, and more. This versatility allows developers to model their data more effectively, choosing the right data structure for the task at hand.

Advantages of Redis Server

- Redis supports atomic operations on these data structures, making it a great fit for scenarios that require consistency and reliability in multi-step operations.

Advantages of Redis Server

- While Redis is an in-memory database, it provides persistence options such as snapshots and append-only files. This allows users to configure the level of durability needed for their specific use case.

Advantages of Redis Server

- Redis supports master-slave replication, enabling the creation of replicas of the master server. This provides high availability and fault tolerance in case the master node fails.

Disadvantages of Redis Server

- Persistence Mechanism Complexity: Redis is an in-memory database, and while it supports persistence, the mechanisms for achieving this (such as snapshots and append-only files) can be complex and may impact performance.
- Redis is an in-memory database, and while it supports persistence, the mechanisms for achieving this (such as snapshots and append-only files) can be complex and may impact performance.
- Limited Query Capability: Redis is not a full-fledged relational database and lacks the complex querying capabilities of traditional databases. It primarily operates on key-value pairs and offers basic data structures like strings, lists, sets, and hashes.
- Redis is not a full-fledged relational database and lacks the complex querying capabilities of traditional databases. It primarily operates on key-value pairs and offers basic data structures like strings, lists, sets, and hashes.
- Memory Usage: Since Redis stores all its data in memory, the amount of data it can handle is limited by the available system memory. Large datasets may require significant memory resources, which can be a potential constraint.
- Since Redis stores all its data in memory, the amount of data it can handle is limited by the available system memory. Large datasets may require significant memory resources, which can be a potential constraint.
- Single-Threaded Nature: Redis traditionally uses a single-threaded event-loop architecture. While this design simplifies certain aspects of the system, it may limit performance on multi-core systems. However, recent versions of Redis have introduced multi-threading in some parts to address this limitation.
- Redis traditionally uses a single-threaded event-loop architecture. While this design simplifies certain aspects of the system, it may limit performance on multi-core systems. However, recent versions of Redis have introduced multi-threading in some parts to address this limitation.
- No Built-in Security Features: Redis initially lacked built-in security features, and it was recommended to be run in trusted environments. While newer versions include authentication

mechanisms, it's essential to configure and manage these security features properly.

- Redis initially lacked built-in security features, and it was recommended to be run in trusted environments.
- While newer versions include authentication mechanisms, it's essential to configure and manage these security features properly.

Disadvantages of Redis Server

- Redis is an in-memory database, and while it supports persistence, the mechanisms for achieving this (such as snapshots and append-only files) can be complex and may impact performance.

Disadvantages of Redis Server

- Redis is not a full-fledged relational database and lacks the complex querying capabilities of traditional databases. It primarily operates on key-value pairs and offers basic data structures like strings, lists, sets, and hashes.

Disadvantages of Redis Server

- Since Redis stores all its data in memory, the amount of data it can handle is limited by the available system memory. Large datasets may require significant memory resources, which can be a potential constraint.

Disadvantages of Redis Server

- Redis traditionally uses a single-threaded event-loop architecture. While this design simplifies certain aspects of the system, it may limit performance on multi-core systems. However, recent versions of Redis have introduced multi-threading in some parts to address this limitation.

Disadvantages of Redis Server

- Redis initially lacked built-in security features, and it was recommended to be run in trusted environments.
- While newer versions include authentication mechanisms, it's essential to configure and manage

these security features properly.

How to Start Redis Server?

To start the Redis server on your machine you need to run the below command in the terminal :

How to Start Redis Server?

```
redis-server
```

How to Start Redis Server?

The above command will start the server on port 6379 which is the default port for the Redis server.

How to Start Redis Server?

You can connect to the redis-server and use it to retrieve and store data using the below command.

How to Start Redis Server?

```
redis-cli
```

Redis Basics

Redis and its role in System Design
How does Redis store data?
Complete Tutorial of Configuration in Redis
Redis Cache

Redis Data Structures

Complete Guide to Redis Lists
Complete Guide on Redis Strings
A Complete Guide to Redis Keys
Complete tutorial on Sets in Redis
Complete tutorial on HyperLogLog in redis
A Complete Guide to Redis Hashes
Complete tutorial on Sorted Sets in Redis

Redis Commands

Complete Guide to Redis Commands
Complete Guide of Redis Scripting
Redis Connections ? Syntax, Command Commands & Examples
Redis ? Client Connection
Complete Guide on Redis Data Types with Commands and Storage
Redis | Publish Subscribe
Difference between Redis Pub/sub vs Redis streams publish
Complete tutorial of Transactions in Redis

Redis Advanced

Understanding Redis PartitioningComplete Guide to Redis PipeliningComplete Guide for Redis BenchmarkComplete Guide to Redis PHPComplete Guide to Redis JavaComplete tutorial on Backup in RedisComplete tutorial on security in Redis

Redis Advanced

Redis Advanced

Feeling lost in the vast world of Backend Development? It's time for a change! Join ourJava Backend Development - Live Courseand embark on an exciting journey to master backend development efficiently and on schedule.What We Offer:

Redis Advanced

- Comprehensive Course
- Expert Guidance for Efficient Learning
- Hands-on Experience with Real-world Projects
- Proven Track Record with 100,000+ Successful Geeks

Redis Advanced

- System Design
- Redis

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers

- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript

- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters

- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing

- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering

- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

SQLite Tutorial

-
-
-

SQLite Tutorial

This SQLite Tutorial is your ultimate guide to mastering this powerful, lightweight database management system. SQLite is widely used in mobile applications, embedded systems, and small to medium-sized websites due to its simplicity and efficiency.

SQLite Tutorial

In this tutorial, we'll walk you through everything you need to know about SQLite, from setting it up and creating your first database to performing complex queries and optimizing performance.

SQLite Tutorial

Whether you're a beginner or an experienced developer, you'll find valuable insights and practical examples to help you make the most of SQLite's features. So, without further ado, let's dive into the tutorial to start learning SQLite.

What is SQLite?

Before exploring the topics SQLite it is very important to understand what is SQLite. So SQLite is a lightweight, self-contained database engine that's easy to use and doesn't require a separate server to operate. It's embedded into applications, making it ideal for mobile apps, small websites, and projects where simplicity and speed are crucial.

What is SQLite?

Unlike other databases, SQLite stores all data in a single file, making it portable and easy to manage. Despite its small size, it supports most SQL standards, allowing developers to perform complex queries and transactions efficiently. SQLite is reliable, fast, and perfect for many everyday database

needs without the overhead of a full-fledged database system.

Features of SQLite

SQLite offers several notable features that make it a popular choice for developers:

Features of SQLite

- Self-contained:SQLite is a complete database system contained in a small library, eliminating the need for a separate server or setup.
- Zero-configuration:It requires no setup or administration, which makes it extremely easy to use right out of the box.
- Serverless:All database interactions are handled directly through function calls in the application, removing the need for a separate database server process.
- Single Database File:All the data, including tables, indexes, and the database schema, is stored in a single cross-platform disk file.
- Reliable and Robust:SQLite is highly reliable with a stable track record, including support forACID(Atomicity, Consistency, Isolation, Durability) transactions.
- Cross-platform:It runs on any platform that supports C, including Windows, macOS, Linux, iOS, and Android.
- Compact:The library size is small, typically around 500KB to 1MB, making it ideal for embedded systems and applications.
- Rich Functionality:Supports most of the SQL92 standard, including complex queries, joins, triggers, and views.
- Public Domain:SQLite is free to use for any purpose, commercial or private, without the need for a license.
- Extensible:Users can add new functions, aggregate functions, and collations at runtime.

Use Case of SQLite

SQLite is versatile and used in a variety of scenarios. Here are some common use cases:

Use Case of SQLite

- Embedded Applications: Ideal for mobile apps, games, and software that need a lightweight, fast database without a server, such as on Android and iOS devices.
- Web Browsers: Used in browsers like Chrome and Firefox to store user data, bookmarks, and settings.
- IoT Devices: Perfect for Internet of Things (IoT) devices that require a compact and efficient database.
- Application File Format: Used as an internal storage format for applications like Adobe Photoshop and Skype to manage configuration settings and user data.
- Standalone Applications: Suitable for desktop applications needing local storage without complex setup, like accounting software or personal finance tools.
- Data Analysis: Handy for storing and querying large datasets during data analysis tasks.
- Testing and Prototyping: Great for testing SQL queries and database designs in a development environment due to its simplicity and ease of use.
- Backup and Archiving: Useful for creating lightweight, portable backups of databases from larger systems.
- Cache for Enterprise Data: Acts as a local cache for enterprise databases, allowing fast access to frequently used data.
- Educational Purposes: Ideal for learning SQL and database management due to its simplicity and accessibility.

SQLite Basics

- SQLite Data Types
- Creating Databases and Tables
- Tables
- Inserting Data into Tables
- Retrieving Data from Tables

- Updating Data in Tables
- Deleting Data from Tables

SQLite Queries and Clauses

- SQLite INSERT
- SQLite SELECT
- SQLite UPDATE
- SQLite Alter Table
- SQLite DELETE
- WHERE Clause
- ORDER BY Clause
- GROUP BY Clause
- HAVING Clause
- SQLite Distinct Clause
- SQLite LIKE Clause
- SQLite GLOB Clause
- SQLite Limit Clause
- Subqueries and Nested Queries
- SQLite UPSERT

SQLite Operators

- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Bitwise Operators
- SQLite BETWEEN
- SQLite IN
- SQLite AND

- SQLite OR
- SQLite IS NULL
- SQLite Union
- SQLite UNION ALL Operator
- SQLite EXCEPT Operator
- SQLite INTERSECT
- SQLite EXISTS
- SQLite Case

SQLite Functions

- SQLite String Functions (LENGTH, SUBSTR, REPLACE, etc.)
- SQLite SUBSTR Function
- SQLite length() Function
- SQLite replace() Function
- Date and Time Functions (DATE, TIME, DATETIME, etc.)
- Date Function
- Mathematical Functions (ABS, ROUND, etc.)
- SQLite abs() Function
- SQLite round() Function

SQLite Aggregate Functions

- SQLite COUNT
- SQLite SUM
- SQLite AVG
- SQLite MIN
- SQLite MAX
- SQLite GROUP_CONCAT
- SQLite TOTAL

SQLite Constraints and Indexes

- Primary Key Constraint
- Unique Constraint
- Not Null Constraint
- Default Constraint
- Check Constraint
- SQLite Index
- SQLite Expression-based Index
- SQLite Foreign Key
- SQLite AUTOINCREMENT

SQLite Joins

- SQLite Join
- SQLite INNER JOIN
- SQLite LEFT JOIN
- SQLite RIGHT JOIN
- SQLite FULL JOIN
- SQLite CROSS JOIN
- SQLite Self-Join

SQLite Transactions

- What are Transactions?
- BEGIN, COMMIT, ROLLBACK Commands
- Savepoints and Rollback to Savepoint

SQLite Triggers

- What are Triggers?
- Creating and Dropping Triggers

- Dropping Triggers
- BEFORE and AFTER Triggers
- INSTEAD OF Triggers

SQLite Views

- What are Views?
- Creating and Dropping Views
- Creating Views
- Dropping Views
- Updating Data through Views

Advanced SQLite Features

- Handling Large Datasets
- Using the ATTACH DATABASE Command
- SQLite - ATTACH Database
- SQLite Show Tables
- SQLite Describe Table
- SQLite Dump
- SQLite Import CSV
- SQLite Export CSV
- SQLite Alter Table
- SQLite Rename Column
- SQLite Drop Table
- SQLite Replace
- How to make recursive query in SQLite?
- Recursive Queries
- Common Table Expressions (CTEs)
- FTS (Full-Text Search)

- Virtual Tables and FTS (Full-Text Search)
- PRAGMA Statements for Database Settings

SQLite Performance Tuning

- Optimizing Queries
- Understanding the Query Execution Plan
- Using ANALYZE for Optimization
- Database Normalization and Denormalization
- Using Transactions Efficiently
- SQLite VACUUM

SQLite Security

- Database Encryption
- User Authentication and Authorization
- Best Practices for Secure Database Handling
- Basic Security Practices for SQLite: Safeguarding Your Data

SQLITE INTERFACES

- SQLite PHP
- SQLite Node.js
- SQLite Java
- SQLite Python
- SQLite Flutter

Conclusion

SQLite is a powerful and easy-to-usedatabase engine, perfect for a wide range of applications. Its simple setup, no need for a server, and small size make it ideal for mobile apps, desktop software, and smart devices.

Conclusion

Despite its lightweight nature, SQLite offers many features that ensure reliable and efficient data management. Whether you're a beginner or an experienced developer, this SQLite tutorial is a great choice for creating fast and dependable applications.

What are the advantages of using SQLite?

SQLite is easy to set up, requires minimal maintenance, and is self-contained within the application. It is widely supported and has a small footprint, making it ideal for mobile development and embedded systems.

Can SQLite handle large databases?

SQLite can handle large databases, but it may not be as efficient as client/server SQL database engines for very large datasets or high concurrency.

How to perform CRUD operations in SQLite?

To perform CRUD (Create, Read, Update, Delete) operations in SQLite, you can use SQL commands such as INSERT, SELECT, UPDATE, and DELETE.

Is SQLite suitable for production applications?

While SQLite is suitable for small to medium-sized applications, it may not be the best choice for large-scale applications due to its file-based nature and limited concurrency support.

Is SQLite suitable for production applications?

Get IBM Certification and a 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Is SQLite suitable for production applications?

Master Data Analysis using Excel, SQL, Python & PowerBI with this complete program and also get a 90% refund. What more motivation do you need? Start the challenge right away!

Is SQLite suitable for production applications?

V

Is SQLite suitable for production applications?

- SQLite
- Databases

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner

- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL

- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development

- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements

- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

What is System Design

- What is Systems Design - Learn System Design
- System Design Life Cycle | SDLC (Design)
- What are the components of System Design?
- Goals and Objectives of System Design
- Why is it Important to Learn System Design?
- Important Key Concepts and Terminologies ? Learn System Design
- Advantages of System Design

System Design Fundamentals

- Analysis of Monolithic and Distributed Systems - Learn System Design
- What is Requirements Gathering Process in System Design?
- Differences between System Analysis and System Design
- Horizontal and Vertical Scaling | System Design
- Capacity Estimation in Systems Design
- Object-Oriented Analysis and Design(OOAD)
- How to Answer a System Design Interview Problem/Question?
- Functional vs. Non Functional Requirements
- Communication Protocols in System Design
- Web Server, Proxies and their role in Designing Systems

Scalability in System Design

- What is Scalability and How to achieve it?
- Which Scalability approach is right for our Application? - System Design
- Primary Bottlenecks that Hurt the Scalability of an Application - System Design

Databases in Designing Systems

- Complete Guide to Database Design - System Design

- SQL vs. NoSQL - Which Database to Choose in System Design?
- File and Database Storage Systems in System Design
- Block, Object, and File Storage in System Design
- Database Sharding - System Design
- Database Replication in System Design

High Level Design(HLD)

- What is High Level Design? ? Learn System Design
- Availability in System Design
- Consistency in System Design
- Reliability in System Design
- CAP Theorem in System Design
- What is API Gateway | System Design?
- What is Content Delivery Network(CDN) in System Design
- What is Load Balancer & How Load Balancing works?
- Caching - System Design Concept
- Communication Protocols in System Design
- Activity Diagrams - Unified Modeling Language (UML)
- Message Queues - System Design

Low Level Design(LLD)

- What is Low Level Design or LLD? - Learn System Design
- Difference between Authentication and Authorization in LLD - System Design
- Performance Optimization Techniques for System Design
- Object-Oriented Analysis and Design(OOAD)
- Data Structures and Algorithms for System Design
- Containerization Architecture in System Design
- Introduction to Modularity and Interfaces In System Design

- Unified Modeling Language (UML) Diagrams
- Data Partitioning Techniques in System Design
- How to Prepare for Low-Level Design Interviews?
- Essential Security Measures in System Design

Design Patterns

- Software Design Patterns Tutorial
- Creational Design Patterns
- Structural Design Patterns
- Behavioral Design Patterns
- Design Patterns Cheat Sheet - When to Use Which Design Pattern?

Interview Guide for System Design

- How to Crack System Design Interview Round?
- Grokking Modern System Design Interview Guide
- System Design Interview Questions and Answers [2024]
- Most Commonly Asked System Design Interview Problems/Questions
- 5 Common System Design Concepts for Interview Preparation
- 5 Tips to Crack Low-Level System Design Interviews

System Design Interview Questions & Answers

- Most Commonly Asked System Design Interview Problems/Questions
- Design Dropbox - A System Design Interview Question
- Designing Twitter - A System Design Interview Question
- System Design Netflix | A Complete Architecture
- System Design of Uber App | Uber System Architecture
- Design BookMyShow - A System Design Interview Question
- Designing Facebook Messenger | System Design Interview

System Design Tutorial

-
-
-

System Design Tutorial

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. This specifically designed System Design tutorial will help you to learn and master System Design concepts in the most efficient way from basics to advanced level.

Importance of System Design in Software Development

Building a successful application goes beyond just having functional features. It is also about ensuring the system can withstand real-world conditions. A well-designed system is not only reliable and scalable but also easy to maintain and evolve as the requirements change. As applications grow in complexity, designing them with scalability, performance, and availability in mind becomes important.

What to Expect from This System Design Tutorial?

This tutorial will guide you through the basic ideas of system design, step by step. Whether you're just starting with system design or want to improve your skills, this guide will help you learn how to create efficient, scalable, and reliable systems. We'll cover everything from the basics of scalability to more advanced techniques like database sharding and caching.

What to Expect from This System Design Tutorial?

By following these steps and continually refining your approach, you can develop robust and efficient system designs that meet both technical requirements and user expectations. For those looking to delve deeper and gain a competitive edge, our System Design course provides a structured pathway to mastering these essential skills.

Basics of System Design

- What is System Design?
- What is the Requirements Gathering Process
- Functional vs Non-Functional Requirements
- Components of System Design
- System Design Life Cycle | SDLC (Design)

Scalability in System Design

- Scalability and How to achieve it
- Horizontal and Vertical Scaling
- Which Scalability approach is right for our Application?
- Primary Bottlenecks that Hurt the Scalability of an Application

System Architectural Styles

- Monolithic Architecture
- Microservices Architecture
- Monolithic vs Microservices Architecture
- Event-Driven Architecture
- Serverless Architecture

High-Level Design(HLD)

- What is High Level Design?
- Availability in System Design
- Consistency in System Design
- Reliability in System Design
- CAP Theorem
- Difference between Concurrency and Parallelism
- Load Balancer

- Consistent Hashing
- Content Delivery Network(CDN)
- Latency and Throughput
- Caching in System Design
- What is API Gateway
- Message Queues
- Communication Protocols
- Network Protocols and Proxies
- Unified Modeling Language (UML)

Databases in Designing Systems

- Which Database to Choose while designing a system ? SQL or NoSQL
- File and Database Storage Systems
- Database Replication in System Design
- Database Sharding
- Block, Object, and File Storage
- Normalization Process in DBMS
- Denormalization in Databases

Low Level Design(LLD)

- What is Low Level Design or LLD
- Object-Oriented Programming(OOP) Concepts
- Data Structures and Algorithms for System Design
- Object-Oriented Analysis and Design
- Difference between Authentication and Authorization
- Design Patterns
- Unit Testing
- Integration Testing

- CI/CD Pipeline
- Introduction to Modularity and Interfaces
- Essential Security Measures in System Design

Interview Questions & Answers of System Design

- URL Shortening Service
- Design Dropbox
- Design Twitter
- System Design Netflix ? Complete Architecture
- System Design of Uber App ? Uber System Architecture
- Design BookMyShow
- Designing Facebook Messenger
- Designing Whatsapp Messenger
- Designing Instagram

Tips for System Design interview

- How to Crack System Design Round in Interviews?
- 5 Tips to Crack Low-Level System Design Interviews
- 5 Common System Design Concepts for Interview Preparation
- 6 Steps To Approach Object-Oriented Design Questions in Interview

Tips for System Design interview

System Design Tutorial

Advantages of System Design

Below are the advantages of System Design:

Advantages of System Design

- In Good system design, it is easy to see how all parts of the system connects, interact and function

together.

- It helps teams plan resources, time, and costs more effectively and reduces any surprises during development.
- An efficient system is easy to maintain and update, which saves time and effort in the long run.
- It allows the system to grow smoothly. You can add new features or handle more users without major changes.
- Good design can enhance the system's performance, making it faster and more efficient.

Disadvantages of System Design

Below are the disadvantages of System Design:

Disadvantages of System Design

- The process of system design can be time-consuming, especially for large and complex systems.
- System design can be expensive, especially if it involves significant research, prototyping, and testing.
- System design requires a significant amount of technical expertise, including knowledge of hardware, software, databases, and data structures.
- Once a system design is complete, it can be difficult to make changes to the design, especially if the design has already been implemented and deployed.

Reference books

Below are some good reference book to learn system design:

Reference books

- ?Design Patterns: Elements of Reusable Object-Oriented Software? by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
- ?The Design of Everyday Things? by Don Norman
- ?Systems Analysis and Design? by Alan Dennis and Barbara Haley Wixom
- ?Clean Architecture: A Craftsman's Guide to Software Structure and Design? by Robert C. Martin

- ?Software Architecture in Practice? by Len Bass, Paul Clements, and Rick Kazman

Reference books

Reference books

Become a System Design expert and also get 90% fee refundon completing 90% course in 90 days!Take the Three 90 Challenge today.

Reference books

Step into the Three 90 Challenge ? 90 days to push limits, break barriers, and become the best version of yourself! Don't miss your chance toupskill and get 90% refund. What more motivation do you need?Start the challenge right away!

Reference books

- System Design
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners

- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025

- GATE DA Course 2025

Similar Reads

Tailwind CSS Layout

- Tailwind CSS Container
- Tailwind CSS Box Sizing
- Tailwind CSS Display
- Tailwind CSS Float
- Tailwind CSS Clear
- Tailwind CSS Object Fit
- Tailwind CSS Object Position
- Tailwind CSS Overflow
- Tailwind CSS overscroll Behavior
- Tailwind CSS Position
- Tailwind CSS Top/Right/Bottom/Left
- Tailwind CSS Visibility
- Tailwind CSS Z-index

Tailwind CSS Flexbox

- Tailwind CSS Flex Direction
- Tailwind CSS Flex Wrap
- Tailwind CSS Flex
- Tailwind CSS Flex Grow
- Tailwind CSS Flex Shrink
- Tailwind CSS Order

Tailwind CSS Grid

- Tailwind CSS Grid Template Columns
- Tailwind CSS Grid Column Start / End
- Tailwind CSS Grid Template Rows
- Tailwind CSS Grid Row Start / End

- Tailwind CSS Grid Auto Flow
- Tailwind CSS Grid Auto Columns
- Tailwind CSS Grid Auto Rows
- Tailwind CSS Gap

Tailwind CSS Alignment

- Tailwind CSS Justify Content
- Tailwind CSS Justify Items
- Tailwind CSS Justify Self
- Tailwind CSS Align Content
- Tailwind CSS Align Items
- Tailwind CSS Align Self
- Tailwind CSS Place Content
- Tailwind CSS Place Items
- Tailwind CSS Place Self

Tailwind CSS Spacing

- Tailwind CSS Padding
- Tailwind CSS Margin
- Tailwind CSS Space Between

Tailwind CSS Sizing

- Tailwind CSS Width
- Tailwind CSS Min-Width
- Tailwind CSS Max-Width
- Tailwind CSS Height
- Tailwind CSS Min-Height
- Tailwind CSS Max-Height

Tailwind CSS Typography

- Tailwind CSS Font Family
- Tailwind CSS Font Size
- Tailwind CSS Font Smoothing
- Tailwind CSS Font Style
- Tailwind CSS Font Weight
- Tailwind CSS Font Variant Numeric
- Tailwind CSS Letter Spacing
- Tailwind CSS Line Height
- Tailwind CSS List Style Type
- Tailwind CSS Placeholder Color
- Tailwind CSS Placeholder Opacity
- Tailwind CSS Text Alignment
- Tailwind CSS Text Color
- Tailwind CSS Text Opacity
- Tailwind CSS Text Decoration
- Tailwind CSS Text Transform
- Tailwind CSS Vertical Alignment
- Tailwind CSS Whitespace
- Tailwind CSS Word Break

Tailwind CSS Backgrounds

- Tailwind CSS Background Image
- Tailwind CSS Background Clip
- Tailwind CSS Background Color
- Tailwind CSS Background Opacity
- Tailwind CSS Background Position

- Tailwind CSS Background Repeat
- Tailwind CSS Background Size
- Tailwind CSS Gradient Color Stops

Tailwind CSS Borders

- Tailwind CSS Border Radius
- Tailwind CSS Border Width
- Tailwind CSS Border Color
- Tailwind CSS Border Opacity
- Tailwind CSS Border Style
- Tailwind CSS Divide Width
- Tailwind CSS Divide Color
- Tailwind CSS Divide Opacity
- Tailwind CSS Divide Style
- Tailwind CSS Ring Width
- Tailwind CSS Ring Color
- Tailwind CSS Ring Opacity
- Tailwind CSS Ring Offset Width
- Tailwind CSS Ring Offset Color

Tailwind CSS Effects

- Tailwind CSS Box Shadow
- Tailwind CSS Opacity

Tailwind Filters

- Tailwind CSS Filter
- Tailwind CSS Blur
- Tailwind CSS Brightness

- Tailwind CSS Contrast
- Tailwind CSS Drop Shadow
- Tailwind CSS Grayscale
- Tailwind CSS Hue Rotate
- Tailwind CSS Invert
- Tailwind CSS Saturate
- Tailwind CSS Sepia
- Tailwind CSS Backdrop Filter
- Tailwind CSS Backdrop Blur
- Tailwind CSS Backdrop Brightness
- Tailwind CSS Backdrop Contrast
- Tailwind CSS Backdrop Grayscale
- Tailwind CSS Backdrop Hue Rotate
- Tailwind CSS Backdrop Invert
- Tailwind CSS Backdrop Opacity
- Tailwind CSS Backdrop Saturate
- Tailwind CSS Backdrop Sepia

Tailwind CSS Tables

- Tailwind CSS Border Collapse
- Tailwind CSS Table Layout

Tailwind CSS Transitions and Animation

- Tailwind CSS Transition Property
- Tailwind CSS Transition Duration
- Tailwind CSS Transition Timing Function
- Tailwind CSS Transition Delay

Tailwind CSS Transforms

- Tailwind CSS Transform
- Tailwind CSS Transform Origin
- Tailwind CSS Scale
- Tailwind CSS Rotate
- Tailwind CSS Translate

Tailwind CSS Interactivity

- Tailwind CSS Appearance
- Tailwind CSS Cursor
- Tailwind CSS Outline
- Tailwind CSS Pointer Events
- Tailwind CSS Resize
- Tailwind CSS User Select

Tailwind CSS SVG

- Tailwind CSS Fill
- Tailwind CSS Stroke
- Tailwind CSS Stroke Width

Tailwind CSS Tutorial

-
-
-

Tailwind CSS Tutorial

Tailwind CSS is a utility-first CSS framework that makes it easy to build custom designs without writing custom CSS. It allows you to apply individual utility classes directly in your HTML, which helps create fully customized layouts with minimal effort.

Tailwind CSS Tutorial

- Tailwind provides many utility classes for building responsive and custom designs.
- You can easily customize the framework to suit your needs via its configuration file.
- Tailwind helps build seamless mobile-first and responsive designs.
- Tailwind CSS has a feature that removes any unused CSS, improving performance.

Tailwind CSS Tutorial

To begin with Tailwind CSS, you need to install it using one of the methods. The easiest way is by using the CDN for quick setup.

Tailwind CSS Tutorial

- Getting Started with Tailwind CSS
- How to Add Tailwind CSS to HTML?
- How to Install Tailwind CSS with Create React App?
- How to setup Tailwind CSS with Vite?
- Install Tailwind CSS with PHP
- Install & Setup Tailwind CSS with Next.js
- How to use Tailwind CSS with Django?
- How to setup Tailwind CSS in AngularJS?

Tailwind CSS Tutorial

Let us now take a look at our first code example.

Tailwind CSS Tutorial

In this example:

Tailwind CSS Tutorial

- Tailwind CDN: The link to the Tailwind CSS CDN is included in the `<head>` section of the HTML document to load the necessary CSS styles.

- Hover Effect: `hover:bg-green-600`: Changes the background color to green when the user hovers over the element. `hover:text-white`: Changes the text color to white when hovered. `transition duration-300 ease-in`: Applies a smooth transition effect for 300ms when hovering.
- `hover:bg-green-600`: Changes the background color to green when the user hovers over the element.
- `hover:text-white`: Changes the text color to white when hovered.
- `transition duration-300 ease-in`: Applies a smooth transition effect for 300ms when hovering.
- Responsive Design: The class names used here are mobile-first, ensuring they work well across different screen sizes.

Tailwind CSS Tutorial

- `hover:bg-green-600`: Changes the background color to green when the user hovers over the element.
- `hover:text-white`: Changes the text color to white when hovered.
- `transition duration-300 ease-in`: Applies a smooth transition effect for 300ms when hovering.

Basics

- Getting Started with Tailwind CSS
- Introduction to Tailwind CSS
- Tailwind CSS: Utility-First Styling for Rapid UI Development

Tailwind CSS Layout

- Container
- Box Sizing
- Display
- Float
- Clear
- Object Fit

- Object Position
- Overflow
- Overscroll Behavior
- Position
- Top/Right/Bottom/Left
- Visibility
- Z-index

Tailwind CSS Flexbox

- Flex Direction
- Flex Wrap
- Flex
- Flex Grow
- Flex Shrink
- Order

Tailwind CSS Grid

- Grid Template Columns
- Grid Column Start / End
- Grid Template Rows
- Grid Row Start / End
- Grid Auto Flow
- Grid Auto Columns
- Grid Auto Rows
- Grid Gap

Tailwind CSS Alignment

- Justify Content

- Justify Items
- Justify Self
- Align Content
- Align Items
- Align Self
- Place Content
- Place Items
- Place Self

Tailwind CSS Spacing

- Padding
- Margin
- Space Between

Tailwind CSS Sizing

- Width
- Min-Width
- Max-Width
- Height
- Min-Height
- Max-Height

Tailwind CSS Typography

- Font Family
- Font Size
- Font Smoothing
- Font Style
- Font Weight

- Font Variant Numeric
- Letter Spacing
- Line Height
- List Style Type
- Placeholder Color
- Placeholder Opacity
- Text Alignment
- Text Color
- Text Opacity
- Text Decoration
- Text Transform
- Vertical Alignment
- Whitespace
- Word Break

Tailwind CSS Backgrounds

- Background Image
- Background Clip
- Background Color
- Background Opacity
- Background Position
- Background Repeat
- Background Size
- Gradient Color Stops

Tailwind CSS Borders

- Border Radius
- Border Width

- Border Color
- Border Opacity
- Border Style
- Divide Width
- Divide Color
- Divide Opacity
- Divide Style
- Ring Width
- Ring Color
- Ring Opacity
- Ring Offset Width
- Ring Offset Color

Tailwind CSS Effects

- Box Shadow
- Opacity

Tailwind CSS Filters

- Filter
- Blur
- Brightness
- Contrast
- Drop Shadow
- Grayscale
- Hue Rotate
- Invert
- Saturate
- Sepia

- Backdrop Filter
- Backdrop Blur
- Backdrop Brightness
- Backdrop Contrast
- Backdrop Grayscale
- Backdrop Hue Rotate
- Backdrop Invert
- Backdrop Opacity
- Backdrop Saturate
- Backdrop Sepia

Tailwind CSS Tables

- Border Collapse
- Table Layout

Tailwind CSS Transitions and Animation

- Transition Property
- Transition Duration
- Transition Timing Function
- Transition Delay

Tailwind CSS Transforms

- Transform
- Transform Origin
- Scale
- Rotate
- Translate

Tailwind CSS Interactivity

- Appearance
- Cursor
- Outline
- Pointer Events
- Resize
- Select

Tailwind CSS SVG

- Fill
- Stroke
- Stroke Width

Tailwind CSS for Interview

- Tailwind CSS Interview Questions and Answers
- Tailwind CSS Exercises, Practice Questions and Solutions

Advantages of Tailwind CSS

- **Faster Development:** Tailwind's utility-first approach allows developers to style elements quickly without writing custom CSS for each element.
- **Customizable:** Tailwind is highly configurable. You can adjust everything, from colors and spacing to breakpoints and fonts, to suit your needs.
- **No Unused CSS:** With tools like PurgeCSS, unused CSS can be removed automatically, keeping your final CSS file size small and optimized.
- **Easier to Maintain:** Tailwind's utility classes encourage consistent design throughout the project, making it easier to maintain.
- **No Need for Custom CSS:** Since you use predefined utility classes, you won't need to write custom CSS for each element.

Applications of Tailwind CSS

- Web Development:Tailwind is perfect for creating custom designs quickly in web applications, websites, and landing pages.
- Responsive Websites:Easily create responsive layouts with Tailwind?s built-in classes for different screen sizes.
- Prototyping:Since Tailwind allows you to style directly in HTML, it?s great for prototyping designs without worrying about CSS.
- Component Libraries:Developers can build a consistent design system by creating reusable components, such as buttons, cards, and navigation bars.

Complete References

- Tailwind CSS Layout
- Tailwind CSS Flexbox
- Tailwind CSS Grid
- Tailwind CSS Alignment
- Tailwind CSS Spacing
- Tailwind CSS Sizing
- Tailwind CSS Typography
- Tailwind CSS Backgrounds
- Tailwind CSS Borders
- Tailwind CSS Effects
- Tailwind CSS Filters
- Tailwind CSS Tables
- Tailwind CSS Transitions and Animation
- Tailwind CSS Transforms
- Tailwind CSS Interactivity

What is Tailwind CSS?

Tailwind CSS is a utility-first CSS framework that provides low-level utility classes to build custom

designs directly in your HTML. It allows for flexible and efficient styling without writing custom CSS.

How to use Tailwind CSS?

To use Tailwind CSS, include it in your project via npm or a CDN. Create a configuration file if needed, then apply utility classes directly to your HTML elements to style them. Tailwind offers a set of pre-defined classes for layout, spacing, typography, and more.

Is Tailwind CSS free?

Yes, Tailwind CSS is free and open-source. You can use it for personal and commercial projects without any cost. It's available under the MIT license.

What is Tailwind CSS used for?

Tailwind CSS is used for designing responsive, customizable user interfaces. It provides utility classes to manage layout, spacing, typography, colors, and more, helping developers create clean and maintainable designs.

Which is better, Tailwind or Bootstrap?

The choice between Tailwind and Bootstrap depends on your needs. Tailwind offers more flexibility with utility-first classes for custom designs, while Bootstrap provides pre-designed components for quicker development. Tailwind is ideal for custom, unique designs, while Bootstrap is better for standardized components.

How to learn Tailwind CSS?

To learn Tailwind CSS, start with this free Tutorial and official documentation. Here, we have created a compiled, easy to learn Tailwind CSS Tutorial.

What are the benefits of using Tailwind CSS?

Tailwind CSS offers several benefits, including faster development with utility classes, a highly customizable design system, and reduced need for custom CSS. It helps maintain consistency and speeds up the UI design process.

Can Tailwind CSS be used with other frameworks?

Yes, Tailwind CSS can be used with other frameworks and libraries. It integrates well with modern JavaScript frameworks like React, Vue, and Angular, and can complement existing styles or components.

Is Tailwind CSS easy to customize?

Yes, Tailwind CSS is highly customizable. You can configure and extend the default design system through the `tailwind.config.js` file, allowing you to tailor the framework to your project's specific needs and branding.

Can Tailwind CSS be used for large projects?

Absolutely. Tailwind CSS is suitable for large projects. Its utility-first approach helps manage complex designs and maintain consistency across extensive codebases. The framework's modularity and customizability make it scalable for various project sizes.

Can Tailwind CSS be used for large projects?

Dive deep into the world of JavaScript with our [JavaScript Full Course Online](#), offering everything from the basics to advanced techniques. Learn how to build interactive websites, optimize your code, and earn a certification to showcase your skills. **Take the 90 Day Challenge!** Complete 90% of the course in 90 days, and earn a 90% refund. Push your limits and stay motivated as you work towards mastering JavaScript. Join now and get started on your JavaScript learning journey!

Can Tailwind CSS be used for large projects?

- CSS
- Tailwind CSS
- Web Technologies
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python

- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP

- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems

- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner

- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming

- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Three.js JavaScript 3D Library

-
-
-

Three.js JavaScript 3D Library

Three.js is an open-source JavaScript library used to display graphics on the web browser. It uses the WebGL API behind the scenes. It allows you to use your GPU (Graphics Processing Unit) to render the graphics and 3D objects on a canvas in the web browser. Since we are using JavaScript, we can also interact with other HTML elements. Three.js was released by Ricardo Cabello in April 2010.

What is Three.js?

Three.js is a JavaScript library designed to simplify the creation of 3D graphics and animations on the web. Acting as a layer of abstraction over the WebGL API, it enables developers to work with 3D graphics in the browser without grappling with the intricacies of low-level WebGL coding. Three.js is a powerful rendering engine, which utilizes WebGL to display 3D graphics.

What is Three.js?

The library employs a scene graph to organize and manage 3D objects hierarchically, incorporating elements like geometries, materials, cameras, and lights. Geometries define shapes, materials determine appearances, and cameras control viewpoints, while lights handle illumination. It also supports animation, enabling developers to update object properties over time, and offers features for applying textures, custom shaders, and user interaction. Overall, Three.js empowers developers to create visually engaging and interactive 3D web applications with ease.

History of Three.js

Three.js, a pioneering JavaScript library, has revolutionized the way developers create and display 3D graphics in web browsers. Here's a comprehensive look at its history, development, and impact on the web development community.

History of Three.js

Three.js was created by Ricardo Cabello, also known as Mr.doob, in April 2010. Ricardo, a talented developer and artist, initiated the project to simplify the process of creating 3D graphics on the web, making it more accessible to developers without deep expertise in graphics programming.

Steps to Install and Setup Three.js

Three.js is a popular JavaScript library used for creating 3D graphics in the browser. Here's a brief overview on how to install and set up Three.js for your project.

Step 1: Setting Up Your Development Environment

Before you start, make sure you have a suitable development environment. You'll need a text editor (like VS Code, Sublime Text, or Atom) and a modern web browser (like Chrome, Firefox, or Edge).

Step 2: Steps to Install and use Three.js

You can install and use Three.js in three ways:

Step 2: Steps to Install and use Three.js

Via CDN Link: You can directly use below CDN link in your project to use Three.js.

Step 2: Steps to Install and use Three.js

Using npm:

Features of Three.js

Three.js is a powerful and versatile JavaScript library for creating 3D graphics on the web. It offers a wide range of features that make it a popular choice for developers looking to build interactive 3D applications. Here are some of the key features of Three.js:

Features of Three.js

- Scene Graph: Three.js provides a robust scene graph, allowing developers to create and manipulate complex 3D scenes with ease. Objects, lights, cameras, and other elements can be

organized hierarchically.

- Geometries:Three.js supports a variety of built-in geometries such as cubes, spheres, cylinders, and toruses. It also allows for the creation of custom geometries, enabling the development of intricate 3D shapes.
- Materials and Shaders:A wide range of materials are available in Three.js, including basic, Lambert, Phong, standard, physical, and more. These materials can be customized with textures, colors, and shaders for advanced visual effects.
- Lighting:Three.js offers various types of lights, such as ambient, directional, point, spot, and hemisphere lights. These lights can be combined and configured to create realistic lighting effects in a scene.
- Textures:Textures can be applied to geometries to enhance their appearance. Three.js supports multiple texture types, including image textures, video textures, and canvas textures. It also supports advanced techniques like bump mapping and normal mapping.
- Animation and Keyframes:Three.js includes robust support for animations. Developers can create keyframe animations, skeletal animations, and morph target animations. The library also supports the interpolation of animations for smooth transitions.
- Cameras:Several types of cameras are available, such as perspective cameras and orthographic cameras. These cameras can be controlled and manipulated to achieve different views and perspectives in a 3D scene.
- Loaders:Three.js provides loaders for various file formats, including OBJ, FBX, GLTF, Collada, and more. This allows developers to import 3D models created in external software and use them in their Three.js projects.
- Post-Processing:Post-processing effects can be applied to scenes to enhance their visual quality. Effects such as bloom, depth of field, and motion blur can be achieved using the post-processing pipeline in Three.js.
- Particles and Sprites:Three.js supports particle systems and sprites, allowing for the creation of effects like smoke, fire, rain, and other particle-based phenomena.

- **Physics Integration:** While Three.js does not have a built-in physics engine, it can be integrated with external physics engines like Cannon.js and Ammo.js to simulate realistic physical interactions and collisions.
- **Cross-Platform Compatibility:** Three.js is compatible with all major web browsers and can be used on various devices, including desktops, tablets, and smartphones. This ensures that 3D applications built with Three.js are accessible to a wide audience.
- **WebXR and VR/AR Support:** Three.js supports WebXR, enabling the development of virtual reality (VR) and augmented reality (AR) applications. This feature allows developers to create immersive experiences that can be accessed through VR headsets and AR devices.

Reason to Learn Three js

- **Web-Based 3D Graphics:** Three.js enables the creation of 3D graphics and animations directly within web browsers, making it an essential tool for web developers looking to enhance user experiences with interactive and visually appealing content.
- **Abstraction Over WebGL:** Three.js abstracts the complexities of WebGL programming, providing a higher-level API that simplifies the process of working with 3D graphics. This allows developers to focus on creativity rather than dealing with low-level WebGL intricacies.
- **Cross-Browser Compatibility:** As a JavaScript library, Three.js is designed to work seamlessly across different web browsers, ensuring that your 3D content is accessible to a wide audience without compatibility issues.
- **Large Community and Resources:** Three.js has a vibrant and active community. Learning and problem-solving become easier with access to a wealth of documentation, tutorials, and community forums where developers share knowledge and solutions.
- **Versatility:** Whether you're creating simple 3D visualizations, games, architectural models, or complex simulations, Three.js is versatile and adaptable to a wide range of applications. Its modular structure allows you to pick and choose components based on your specific needs.
- **Rapid Prototyping:** Three.js facilitates rapid prototyping of 3D projects, allowing developers to

quickly experiment with ideas and iterate on designs without the steep learning curve associated with lower-level graphics programming.

Advantages of Three.js

- **WebGL Abstraction:**Three.js abstracts the complexities of WebGL, providing a higher-level API that simplifies the creation of 3D graphics on the web.
- **Cross-Browser Compatibility:**Three.js is designed to work across different web browsers, ensuring consistent performance and visual quality for a broad audience.
- **Community Support:**Three.js has a large and active community, offering a wealth of resources, tutorials, and forums for developers to learn and seek assistance.
- **Rapid Development:**The library facilitates rapid prototyping and development of 3D projects, allowing developers to quickly experiment with ideas and iterate on designs.
- **Versatility:**Three.js is versatile and applicable to a wide range of 3D applications, from simple visualizations to complex simulations and games.
- **Rich Feature Set:**The library comes with a comprehensive set of features, including built-in geometries, materials, cameras, lights, and animation support.
- **Performance Optimization:**Three.js includes features for optimizing performance, such as frustum culling and level-of-detail support, ensuring smooth rendering even in complex scenes.
- **Compatibility with Other Tools:**Three.js can be integrated with other popular tools and frameworks, providing flexibility and compatibility with technologies like React, Angular, and more.
- **Virtual and Augmented Reality (VR/AR):**Three.js supports the development of virtual and augmented reality experiences, making it suitable for emerging technologies and immersive web content.
- **Open Source and Active Development:**Being an open-source project, Three.js benefits from continuous development and updates, keeping it in line with the latest web standards and technologies.

How do I get started with Three.js?

To get started with Three.js:

How do I get started with Three.js?

- Include Three.js: Add the Three.js library to your HTML file using a CDN or download it locally.
- Create a Scene: Set up a scene, camera, renderer, and objects like geometries and materials.
- Animate and Render: Create an animation loop to update the scene and render it using the renderer.

Is Three.js suitable for beginners?

Three.js offers a high-level API that abstracts many complexities of WebGL, making it accessible for beginners to start creating 3D graphics on the web. However, familiarity with JavaScript and some understanding of 3D concepts (such as coordinates, geometries, and materials) will be beneficial.

How does Three.js Handle User Interaction?

Three.js enables user interaction by responding to events such as mouse clicks and movements. Developers can use this feature to create interactive interfaces having 3D look.

What is a Geometry in Three.js?

In Three.js, a geometry defines the shape of a 3D object, such as a cube or sphere. We use this to create an interactive UI. It specifies the vertices, edges, and faces that make up the object.

Why is the Scene Graph Important in Three.js?

Because it is important for organizing and managing the hierarchy of 3D objects in a scene. It helps developers to structure and control the relationships between different elements so that they can build unique graphs.

How does Three.js Support Animation?

Three.js supports animation by allowing developers to update the properties of 3D objects over time. It allows changes in position, rotation, and scale, providing the basis for dynamic and engaging visual experiences. These all features make animation possible.

How can Three.js Contribute to Responsive Design?

Three.js supports responsive design by allowing developers to create 3D content that adapts well to different screen sizes and resolutions, ensuring a consistent user experience across devices. That's why it is highly used in creation of interactive 3D UI.

What are some examples of applications built with Three.js?

Three.js is used in a wide range of applications, including:

What are some examples of applications built with Three.js?

- Games and Interactive Experiences: Virtual reality (VR) games, simulations, and interactive websites.
- Data Visualization: Complex data visualizations and scientific simulations.
- Architectural and Product Visualization: Virtual tours, architectural walkthroughs, and product configurators.
- Education and Training: Interactive educational tools and training simulations.

What are some examples of applications built with Three.js?

Master DSA with JavaScript in just 90 days. Explore core DSA concepts, refine your coding skills, and tackle real-world challenges. Take on the Three 90 Challenge? complete 90% of the course in 90 days and earn a 90% refund as a reward for your commitment!

What are some examples of applications built with Three.js?

M

What are some examples of applications built with Three.js?

- JavaScript
- Web Technologies
- Three.js
- Web-Tech Tutorials

- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy

- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes

- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel

- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation

- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Basics

- Vue.js Introduction & Installation
- Vue.js Instances
- Vue.js Watchers
- Vue.js Methods
- Vue.js Event Modifiers
- Vue.js DOM tree
- How to write and use for loop in Vue js ?
- Vue.js Two Way Binding Model
- Vue.js Reusing Components
- Vue.js List Rendering
- Vue.js List Rendering Mutation Methods
- Vue.js v-cloak Directive
- Vue.js Passing Data to Child Components with Props
- Vue.js Form Input Binding with Select option
- Vue.js Dynamic Components
- Vue.js Form Input Value Binding
- Vue.js Form Input Binding number Modifier
- Vue.js List Rendering v-for with v-if
- Vue.js List Rendering v-for with a Range
- Vue.js Form Input Binding with Checkbox option
- Vue.js Form Input Binding with Multiline text
- Vue.js Form Input Binding trim Modifier
- Vue.js Form Input Binding with Radio Option
- Vue.js List Rendering v-for with an Object
- Vue.js Render Function with h() Arguments
- Vue.js Composition API with Templates

- Vue.js Event Handling
- Vue.js Declarative Rendering
- Create a Hover effect in Vue.js
- Types of data binding with template in Vue.js
- Vue.js Click Event
- Pass data between components using Vue.js Event Bus
- Vue.js Render Functions Component VNodes creation
- Vue.js List Entering & Leaving Transitions
- Vue.js Composition API using Provide
- Vue.js List Move Transitions
- Vue.js Transitioning between the Components
- REST API Call to Get Location Details in Vue.js

Transition

- Vue.js v-on:keyup Directive
- v-bind Directive in Vue.js
- v-for Directive in Vue.js

Directives

- Vue.js Conditional Rendering
- Vue.js List Rendering v-for with a Component
- Vue.js List Rendering v-for on a <template>
- Vue.js | v-if directive
- Vue.js | v-text directive
- Vue.js | v-show directive
- Vue.js | v-html directive
- Vue.js v-on:click directive
- Vue.js v-once Directive

- Vue.js v-on:click.ctrl Directive
- Vue.js v-on:click.right Directive
- Vue.js v-on:click.shift Directive
- Vue.js v-on:click.left Directive
- Vue.js v-on:click.alt Directive
- How a View-model works in Vue.js?
- v-on:click.middle Directive in Vue.js
- Vue.js v-pre Directive
- Vue.js Form Input Binding lazy Modifier
- Data Conversion to KB, MB, GB, TB using Vue.js filters
- Convert decimal point numbers to percentage using filters in Vue.js
- Capitalizing a string using filters in VueJS
- Vue.js Placeholder using filters
- Truncate a String using filter in Vue.js

Filters

- Vue.js | Routing
- How to use routing in Vue.js ?
- Vue.js Vuex
- Consuming a Rest API with Axios in Vue.js

Routing

- What is the difference between ShadowDOM and VirtualDOM ?
- What is the difference between created and mounted event in VueJS?

Difference Between

Vue.js Tutorial

-

-

-

Vue.js Tutorial

Vue.js is a progressive JavaScript framework for building user interfaces. It stands out for its simplicity, seamless integration with other libraries, and reactive data binding.

Vue.js Tutorial

- Built on JavaScript for flexible and component-based development.
- Supports declarative rendering, reactivity, and two-way data binding.
- Provides a lightweight framework with powerful features for single-page applications (SPAs).
- Easy integration into existing projects or libraries.

Vue.js Tutorial

To start with Vue.js, you need to install or set it up on your machine. Follow these articles to install depending on your system:

Vue.js Tutorial

- [Vue.js Introduction & Installation](#)
- [Getting Started with VueJS](#)

Vue.js Tutorial

Let us now take a look at our first code example.

Vue.js Tutorial

It will start a server, and when you visit <http://localhost:8080>, it will display

Vue.js Tutorial

In this example:

Vue.js Tutorial

- The App.vue file defines the structure, logic, and styling of the application in a single file component.
- message is a reactive property defined in the data() function and rendered in the template using {{ }}.
- The application is mounted to the #app div in index.html (automatically generated in the Vue project setup).

Why Learn Vue.js?

- Lightweight and easy to integrate with existing projects.
- Offers excellent performance for small- to medium-sized applications.
- Supports two-way data binding and reactivity.
- Backed by a robust community and extensive documentation.

Why Learn Vue.js?

Vue.js Tutorial Prerequisites:HTML,CSSandJavaScript

Vue.js Basics

- Introduction & Installation
- Instances
- Event Modifiers
- DOM tree
- How to write and use for loop in Vue js ?
- Two Way Binding Model
- Reusing Components
- Rendering
- List Rendering Mutation Methods
- v-cloak Directive
- Passing Data to Child Components with Props

- Form Input Binding with Select option
- Dynamic Components
- Form Input Value Binding
- Form Input Binding number Modifier
- List Rendering v-for with v-if
- List Rendering v-for with a Range
- Form Input Binding with Checkbox option
- Form Input Binding with Multiline text
- Form Input Binding trim Modifier
- Form Input Binding with Radio Option
- List Rendering v-for with an Object
- Render Function with h() Arguments
- Form Input Binding with Radio Option
- List Rendering v-for with an Object
- Composition API with Templates
- Event Handling
- Event Modifiers
- Declarative Rendering
- Create a Hover effect in Vue.js
- Types of data binding with template in Vue.js
- Click Event
- Dynamic Components
- Pass data between components using Vue.js Event Bus
- Passing Data to Child Components with Props
- Render Functions Component VNodes creation
- List Entering & Leaving Transitions
- Composition API using Provide

- Watchers

- Methods

Vue.js Directives

- v-on:keyup Directive

- v-bind Directive in Vue.js

- v-for Directive in Vue.js

- List Rendering v-for with a Range

- v-else-if Directive in Vue.js

- Custom Directives with Components

- Form Input Binding number Modifier

- Conditional Rendering

- List Rendering v-for with a Component

- List Rendering v-for on a <template>

- v-if directive

- v-text directive

- v-show directive

- v-html directive

- v-on:click directive

- v-once Directive

- v-on:click.ctrl Directive

- v-on:click.right Directive

- v-on:click.shift Directive

- v-on:click.left Directive

- v-on:click.middle Directive in Vue.js

- v-pre Directive

Vue.js Transition

- List Move Transitions
- Transitioning between the Components
- REST API Call to Get Location Details

Vue.js Routing

- Routing
- How to use routing in Vue.js ?

Advanced Vue

- Consuming a Rest API with Axios in Vue.js

Differences

- Node.js vs Vue.js
- What is the difference between ShadowDOM and VirtualDOM ?
- What is the difference between created and mounted event in VueJS?
- Comparative study of Svelte vs React vs Angular vs Vue
- What is the difference between one-way data flow and two-way data binding in vue.js?

Miscellaneous

- How to replace all occurrences of a string using Vue.js filters ?
- How to set a click event once a page or view is loaded in vue.js?
- How to check an image is loaded or not in VueJS ?
- How to loop through a list of elements and display it in VueJS ?
- How to Make localStorage Reactive in Vue.js ?
- How to implement DateTime localization in vue.js ?
- Select different values on two select tags in Vue.js
- How to convert a normal string to a uppercase string using filter in VueJS ?
- How to dynamically add or remove items from a list in Vue.js ?
- How to binding inline styles in Vue.js ?

- How to change text into lower case string using filters in Vue.js ?
- How to Add Custom Fonts in VueJS ?
- How to Trigger Watchers on Initialization in Vue.js ?
- How to replace jQuery with Vue.js ?
- How to catch props data on component change in Vue.js ?
- How to create a reporting app with Vue 3 and Composition API ?
- How to delete all occurrences of a particular string using Vue.js ?
- How to Toggle a Class in Vue.js ?
- How to Check Response Data Length in VueJS ?
- Why the component data must be a function in vue.js ?

Vue.js for Interview

- VueJS Interview Questions and Answers

Applications of Vue.js

- Single-Page Applications (SPAs): Vue.js is an excellent choice for building dynamic and interactive SPAs.
- User Interfaces (UIs): Vue.js is widely used for creating engaging and responsive UIs.
- Real-Time Updates: Vue.js provides real-time data updates, making it suitable for applications that require live data synchronization.
- Prototyping: Due to its simplicity and flexibility, Vue.js is perfect for rapid prototyping of web applications.
- Cross-Platform Development: Whether you're building web, mobile, or desktop applications, Vue.js can adapt to different platforms seamlessly.

Vue.js vs Other Frameworks

Feature | Vue.js | React | Angular

Syntax | Simple and declarative | JSX-based | TypeScript-based

Learning Curve | Easy | Moderate | Steep

Performance | High | High | Moderate

Ecosystem | Vuex, Vue Router | Redux, React Router | Built-in tools

Use Case | SPAs, Prototypes | Large-scale apps | Enterprise-grade apps

Vue.js vs Other Frameworks

Master DSA with JavaScript in just 90 days. Explore core DSA concepts, refine your coding skills, and tackle real-world challenges. Take on the Three 90 Challenge?complete 90% of the course in 90 days and earn a 90% refund as a reward for your commitment!

Vue.js vs Other Frameworks

M

Vue.js vs Other Frameworks

- JavaScript
- Web Technologies
- Vue.JS

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution

- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures

- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs

- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker

- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy

- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced

- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course

- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE

- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

CSS Frameworks

- Bootstrap Tutorial
- Bootstrap 5 Tutorial
- Tailwind CSS Tutorial
- Bulma Tutorial
- Pure CSS
- Materialize CSS
- Foundation CSS Tutorial
- Semantic UI
- Primer CSS
- Spectre CSS

CSS Preprocessor

- SASS
- LESS

JavaScript Libraries

- Lodash
- D3.js
- Vue.js Tutorial
- Underscore.js
- TensorFlow.js
- Collect.js
- Fabric.js
- p5.js

ReactJS

- React Tutorial

- Next.js Tutorial
- React Material UI
- React Bootstrap Tutorial
- React Suite
- Ant Design
- ReactJS Reactstrap
- BlueprintJS

AngularJS

- AngularJS Tutorial
- Angular PrimeNG
- Angular ngx Bootstrap

jQuery

- jQuery Tutorial
- jQuery UI
- jQuery Mobile
- jQWidgets

Web Development

-
-
-

Web Development

Web development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites. Web development is typically broken down into 3 core areas: Frontend, Backend, Full Stack Development.

Frontend Development

The Face of the Web. The part of a website where the user interacts directly is termed as front end. This involves designing the structure, layout, and behavior of the website. It is also referred to as the ?client side? of the application.

Frontend Technologies

- HTML:HTML stands for HyperText Markup Language. It is used to design the front end portion of web pages using markup language. It acts as a skeleton for a website since it is used to make the structure of a website.
- CSS:Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. It is used to style our website.
- JavaScript:JavaScript is a scripting language used to provide a dynamic behavior to our website.
- Bootstrap:Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular CSS framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).Bootstrap 4Bootstrap 5
- Bootstrap 4
- Bootstrap 5

Frontend Technologies

- Bootstrap 4
- Bootstrap 5

Frontend Frameworks/Libraries

- React.js: A popular JavaScript library for building dynamic, component-based user interfaces.
- Angular: A full-fledged framework for building single-page applications (SPAs), with features like two-way data binding and dependency injection.
- Vue.js: A progressive JavaScript framework that is flexible and can be used for building both

simple and complex user interfaces.

Backend Development

The Backbone of the Web. Backend is the server side of a website. It is part of the website that users cannot see and interact with. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

Server-side Programming Languages and Frameworks

- PHP:PHP is a server-side scripting language designed specifically for web development.
- Java:Java is one of the most popular and widely used programming languages. It is highly scalable.
- Python:Python is a programming language that lets you work quickly and integrate systems more efficiently.
- Node.js:Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside a browser.
- Ruby:Ruby is a dynamic, reflective, object-oriented, general-purpose programming language.
- C#:C# is a high-level, general-purpose programming language developed by Microsoft.

Server-side Programming Languages and Frameworks

Backend Languages | Backend Frameworks

PHP | Laravel, Wordpress

Java | Spring, Hibernate

Python | Django, Flask, Python PIP

Node.js | Express

Ruby | Ruby on Rails

C# | .NET

Server-side Programming Languages and Frameworks

Backend Languages

Server-side Programming Languages and Frameworks

Backend Frameworks

Server-side Programming Languages and Frameworks

PHP

Server-side Programming Languages and Frameworks

Laravel, Wordpress

Server-side Programming Languages and Frameworks

Java

Server-side Programming Languages and Frameworks

Spring, Hibernate

Server-side Programming Languages and Frameworks

Python

Server-side Programming Languages and Frameworks

Django, Flask, Python PIP

Server-side Programming Languages and Frameworks

Node.js

Server-side Programming Languages and Frameworks

Express

Server-side Programming Languages and Frameworks

Ruby

Server-side Programming Languages and Frameworks

Ruby on Rails

Server-side Programming Languages and Frameworks

C#

Server-side Programming Languages and Frameworks

.NET

Databases

- MySQL
- PostgreSQL
- MongoDB
- MariaDB
- SQLite

APIs (Application Programming Interfaces)

- RESTful API's
- GraphQL

Full Stack Development

Full-stack development refers to the practice of developing both the frontend and backend of a website or web application. Full-stack developers have a deep understanding of both areas and can build end-to-end solutions.

Full Stack Development

Full Stack Technologies:

- MERN Stack: MongoDB, Express.js, React, Node.js
- MEAN Stack: MongoDB, Express.js, Angular, Node.js
- JAMstack: JavaScript, APIs, Markup
- Django Stack: Django, MySQL/PostgreSQL, HTML/CSS/JavaScript
- Spring Boot Stack: Spring Boot, MySQL/PostgreSQL, Java

- LAMP Stack: Linux, Apache, MySQL, PHP
- LEMP Stack: Linux, Engine-X, MySQL, PHP

Full Stack Technologies:

Databases

In web technology, a database is a structured collection of data that is stored electronically and accessed via a web application. It serves as the backend component where data is stored, managed, and retrieved. Databases can be relational (like MySQL, PostgreSQL) using structured tables and SQL for queries, or non-relational (like MongoDB, CouchDB) which store data in flexible, document-oriented formats. They enable web applications to handle dynamic content, user data, transactions, and more by providing efficient storage, retrieval, and manipulation capabilities. Database management systems (DBMS) are used to interact with the database, ensuring data integrity, security, and performance.

1. Relational Databases

A relational database stores data in tables, similar to a spreadsheet, where each table has rows and columns. The rows hold individual records, and the columns define the data attributes. Tables can be linked to each other through special keys, allowing related data to be connected.

1. Relational Databases

- PostgreSQL: PostgreSQL is a powerful, open-source relational database that supports advanced SQL features and complex queries. It handles structured data, ensures ACID compliance, and is known for its reliability and extensibility.
- MariaDB: MariaDB is an open-source relational database that evolved from MySQL, offering improved performance, security, and features. It supports SQL queries, ACID compliance, and is highly compatible with MySQL.
- MySQL: MySQL is an open-source relational database management system that uses SQL for managing structured data. It's known for its reliability, ease of use, and performance, widely used in

web applications.

2. NoSQL Databases

A NoSQL database stores data in a flexible, non-tabular format, unlike traditional relational databases. Instead of using tables with rows and columns, NoSQL databases might use documents, key-value pairs, wide-columns, or graphs to store data. This allows them to handle large amounts of unstructured or semi-structured data efficiently. They are designed to scale easily and manage big data applications.

2. NoSQL Databases

- MongoDB: MongoDB is a NoSQL database storing data in JSON-like documents. It handles unstructured data, supports powerful queries, and scales easily across servers, making it popular for flexible, scalable applications.
- Cassandra: Apache Cassandra is an open-source NoSQL database that is used for handling big data. It has the capability to handle structure, semi-structured, and unstructured data.
- Redis: Redis is an in-memory NoSQL database known for its speed. It supports various data structures like strings, hashes, and lists, making it ideal for caching, real-time analytics, and messaging.

2. NoSQL Databases

Must Read :

2. NoSQL Databases

- Top 10 Tools for Modern Web Development
- Top 10 Frameworks for Web Applications
- 7 Best Languages for Web Development

2. NoSQL Databases

Become a Full-Stack Developer and also get 90% fee refundon completing 90% course in 90

days!Take the Three 90 Challenge today.

2. NoSQL Databases

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need?Start the challenge right away!

2. NoSQL Databases

- Web Technologies
- GBlog
- Tutorials
- GBlog 2025

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge

- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions

- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance

- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk

- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article

- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]

- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Blockchain Technology

- Introduction to Blockchain technology | Set 1
- History of Blockchain
- Features of Blockchain
- Important Blockchain terminologies
- Different Version of Blockchain
- Types of Blockchain
- Public Blockchain
- An Introduction to Private Blockchain
- Hybrid Blockchain
- What is Consortium Blockchain?
- Top Applications of Blockchain in the Real World
- Advantages and Disadvantages of Blockchain
- Benefits of Blockchain Technology
- How does the Blockchain Work?
- What is P2P (Peer-to-Peer Process)?
- What is Decentralization? Definition, Working, Need, Benefits
- Components of Blockchain Network
- Centralized vs. Decentralized vs. Distributed Systems
- Difference between Public and Private blockchain

Consensus Algorithms

- Consensus Algorithms in Blockchain
- Blockchain - Proof of Work (PoW)
- Proof of Burn (PoB) Consensus Algorithm in Blockchain
- Proof of Stake (PoS) in Blockchain
- Byzantine Generals Problem in Blockchain

- Cryptographic Consensus Mechanisms in Blockchain
- Delegated Proof Of Stake (DPoS)
- practical Byzantine Fault Tolerance(pBFT)

Blockchain Architecture

- Blockchain Structure
- Candidate block in Blockchain
- Actors Involved in Blockchain Solution
- Blockchain Transaction Lifecycle
- Blockchain Forks
- Blockchain and Block Header
- Blockchain Incentives to Miners
- Core Component of Blockchain
- Blockchain Protocols and Their Working
- What is Blockchain Authentication?

Blockchain and Cryptocurrency

- What is Cryptocurrency?
- Advantages and Disadvantages of Cryptocurrency in 2020
- How are Cryptocurrencies Created?
- What is a Cryptographic Token?
- What is Cryptoeconomics?
- What is an Initial Coin Offering (ICO)?
- Generalized Proof-of-Stake Mining in Cryptocurrencies
- Blockchain - Electronic Cash
- What is Blockchain Wallet?

Smart Contracts and Decentralized Applications (DApps)

- What is Ethereum: Working, Types, Features
- Components of Ethereum Network
- Difference Between Bitcoin and Ethereum
- What are Ethereum Accounts?
- What are Nodes and Clients in Ethereum?
- What is Ethereum Virtual Machine and How it Works?
- Ethereum - Gas and Fees
- How to Simply Deploy a Smart Contract on Ethereum?
- "Hello World" Smart Contract in Remix-IDE
- What are Decentralized Apps (dApps) in Blockchain?
- Pros, Cons, and Examples of dApp
- DAO(Decentralized Autonomous Organization) in Blockchain

Blockchain Security

- Blockchain and Data Privacy
- Blockchain and KYC
- Role of Blockchain in Cybersecurity
- Blockchain to Secure IoT Data

Blockchain Applications

- Applications of Blockchain in Data Management
- Benefits and Applications of Blockchain in Cloud Computing
- Integration of Artificial Intelligence and Blockchain
- How Blockchain Can Change the Future of Banking?
- Blockchain - Into the Future
- Blockchain in Genomics
- Integration of Blockchain and AI
- Use Cases of Blockchain in different fields

- Role of Blockchain in Sustainable Development
- Applications and Uses of Blockchain
- Benefits of Blockchain in Healthcare
- Decentralized Voting System using Blockchain

Blockchain Implementation

- Create simple Blockchain using Python
- Implementation of Blockchain in Java
- Build a To-do List Web Application Powered by Blockchain
- Flutter and Blockchain - Hello World Dapp
- Blockchain Gaming : Part 1 (Introduction)
- How to use GANACHE Truffle Suite to Deploy a Smart Contract in Solidity (Blockchain)?
- How to use MetaMask to Deploy a Smart contract in Solidity (Blockchain)?
- Build a Authentication Using Blockchain

Miscellaneous

- Difference between Blockchain and a Database
- Difference between Proof of Work (PoW) and Proof of Stake (PoS) in blockchain
- Difference Between DAG and Blockchain
- Bitcoin vs Cryptocurrency
- Full vs Simple Payment Verification in Blockchain
- Difference Between Bitcoin and Blockchain
- Hyperledger vs Ethereum in Blockchain
- Difference Between Blockchain and Hashgraph
- Quorum Blockchain
- What is EOS Blockchain?
- What's the connection between Java and Blockchain?
- Top 10 Programming Languages for Blockchain Development

- How does BlockChain support Crowdfunding ?
- Blockchain Interoperability
- Why Blockchain is Impenetrable?
- Different job roles in Blockchain Technology
- Demur-rage currencies in Blockchain
- Blockchain of Things (BoT)
- What is Blockchain Ecosystem?
- CAP Theorem in Blockchain

Interview Questions

- Top 50 Blockchain Interview Questions and Answers
- 15 Top Blockchain Interview Questions & Answers 2024

Blockchain Tutorial

-
-
-

Blockchain Tutorial

Blockchain technology in simple words is a digital database where information or data is stored in blocks that are linked together to form a chain. This Blockchain Tutorial covers all basic to advanced topics of blockchain like cryptography, Blockchain Algorithms, Blockchain Architecture, Blockchain Security, Smart Contracts, etc.

Blockchain Tutorial

Blockchain Tutorial

Table of Content

Blockchain Tutorial

- What is Blockchain?
- Why Do We Use Blockchain?
- Introduction to Blockchain Technology
- Cryptography
- Consensus Algorithms
- Blockchain Architecture
- Blockchain and Cryptocurrency
- Smart Contracts and Decentralized Applications (DApps)
- Blockchain Security
- Blockchain Applications
- Blockchain Implementation
- Miscellaneous
- Interview Questions

What is Blockchain?

A Blockchain is a digital ledger of transactions that is secure, immutable, and decentralized. It consists of a chain of blocks and each block contains a secure group of transactions of money, bitcoins, contracts, property, etc. without any need for third-party middle-man such as central authorities, banks, government, etc. Transactions are verified by a network of computers. Once a block of information is created in the chain, it can't be changed or deleted. This makes the blockchain very secure and trustworthy. It is a software protocol, it never runs without the internet.

What is Blockchain?

A Blockchain is a chain of blocks that contains information about transactions. On completion of a transaction, it will go to the blockchain's permanent database. It is mainly used for secure transactions without any third-party involvement in between. To visualize blocks, transactions, and blockchain network metrics, you can use blockchain explorer.

Why Do We Use Blockchain?

Blockchain is a distributed digital ledger of transactions. It changes the whole concept of the transaction now we don't have any need for centralized authorities to verify our transactions, blockchain makes it decentralized and secure. The security is maintained by cryptographic processes performed by p2p users, through the process called Data mining. Here are some important features given below:

Why Do We Use Blockchain?

- Secure
- Transparent
- Tamper-proof
- Peer-to-peer transaction

Introduction to Blockchain Technology

- Introduction to Blockchain Technology
- History of Blockchain
- Features of Blockchain
- Important Blockchain Terminologies
- Different Versions of Blockchain
- Types of Blockchain
- Public Blockchain
- Private Blockchain
- Hybrid Blockchain
- Consortium Blockchain
- Applications of Blockchain
- Advantages and Disadvantages of Blockchain
- Benefits of Blockchain Technology
- How Does the Blockchain Work?

- Need of Blockchain
- Peer-to-Peer Network
- What is Decentralization?
- What is Ledger?
- What is Blockchain Distributed Ledger?
- Difference between Centralized and Distributed Ledgers
- Difference between Public and Private Ledgers

Cryptography

- Cryptography and its Types
- Cryptography in Blockchain
- Blockchain Hash Function
- Blockchain ? Block hashing
- How Block Hashes Work in Blockchain
- What is Digital Signature?
- Blockchain ? Public Key Cryptography
- Blockchain ? Private Key Cryptography
- Blockchain Merkle Trees
- What is Cryptographic Primitive in Blockchain?
- Blockchain ? Elliptic Curve Cryptography
- How to Create a Public/Private Key Pair?
- Blockchain ? Encrypt & Decrypt Files With Password Using OpenSSL
- Blockchain ? Elliptic Curve Digital Signature Algorithm (ECDSA)
- Blockchain Resolving Conflicts
- Applications of Homomorphic Encryption in Blockchain

Consensus Algorithms

- Consensus Algorithms in Blockchain

- Blockchain Proof Of Work
- Proof of Work Consensus algorithm
- Proof of Burn Consensus Algorithm in Blockchain
- Proof of Stake (PoS) in Blockchain
- Byzantine Generals Problem in Blockchain
- Cryptographic Consensus Mechanisms in Blockchain
- Delegated Proof Of Stake (DPoS)
- practical Byzantine Fault Tolerance(pBFT)

Blockchain Architecture

- Blockchain Structure
- Genesis Block in Blockchain
- Candidate Block in Blockchain
- Actors Involved in Blockchain Solution
- Blockchain Transaction Life-cycle
- Blockchain Forks
- Blockchain and Block Header
- Components of Blockchain Network
- Blockchain Incentives to Miners
- Core Component of Blockchain
- Blockchain Protocols and Their Working
- Blockchain ? Resolving Conflicts
- What is Blockchain Authentication?

Blockchain and Cryptocurrency

- What is Blockchain Cryptocurrency?
- Cryptocurrencies Benefits
- How are Cryptocurrencies created

- What is a Cryptographic Token
- Non-Fungible Tokens (NFTs)
- What is Cryptoeconomics?
- What is an ICO?
- Generalized Proof-of-Stake Mining in Cryptocurrencies
- Blockchain ? Electronic Cash
- What is Blockchain Wallet and How does it work

Smart Contracts and Decentralized Applications (DApps)

- What is Ethereum
- Components of the Ethereum Network
- Difference between Ethereum and Bitcoin
- What is Ether
- Ethereum Accounts
- Nodes and Clients in Ethereum
- What is Ethereum Virtual Machine
- Ethereum ? Gas and Fees
- What is a Smart Contract
- How to Test a Smart Contract for Ethereum?
- How to Simply Deploy a Smart Contract on Ethereum
- Hello World ? Smart Contract in Remix-IDE
- Decentralized Apps (dApps) in Blockchain
- What is an Ethereum API
- Pros and Cons of Dapps
- What is a Dao?

Blockchain Security

- Blockchain and Data Privacy

- Blockchain and KYC
- Role of Blockchain in Cybersecurity
- Blockchain to Secure IoT Data

Blockchain Applications

- Applications of Blockchain in Data Management
- Top Applications of Blockchain in the Real World
- Benefits and Applications of Blockchain in Cloud Computing
- Integration of Artificial Intelligence and Blockchain
- How Blockchain Can Change the Future of Banking?
- Blockchain ? Into the Future
- Blockchain in Genomics
- Integration of Blockchain and AI
- Use Cases of Blockchain in different fields
- Role of Blockchain in Sustainable Development
- Applications and Uses of Blockchain
- Benefits of Blockchain in Healthcare
- Decentralized Voting System Using Blockchain

Blockchain Implementation

- Create a simple Blockchain using Python
- Implementation of Blockchain in Java
- Build a To-do List Web Application Powered by Blockchain
- Flutter and Blockchain ? Hello World Dapp
- Blockchain Gaming: Part 1 (Introduction)
- How to Use GANACHE Truffle Suite to Deploy a Smart Contract in Solidity (Blockchain)?
- How to Use MetaMask to Deploy a Smart Contract in Solidity (Blockchain)?
- Build an Authentication Using the Blockchain

Miscellaneous

- Public Blockchain vs Private Blockchain
- Blockchain vs Database
- Proof of Work (PoW) vs Proof of Stake (PoS)
- DAG vs Blockchain
- Hot Wallets vs Cold Wallets in Blockchain
- Bitcoin vs Cryptocurrency
- Full Payment Verification vs Simple Payment Verification in Blockchain
- Bitcoin vs Blockchain
- Hyperledger vs Ethereum
- Blockchain vs Hashgraph
- Quorum Blockchain
- What is EOS Blockchain?
- DAO(Decentralized Autonomous Organization) in Blockchain
- What's the connection between Java and Blockchain?
- Top 10 Programming Languages for Blockchain Development
- How does Blockchain support Crowdfunding?
- Blockchain Interoperability
- Why Blockchain is Impenetrable?
- Different job roles in Blockchain Technology
- Demerage currencies in Blockchain
- Blockchain of Things (BoT)
- What is Blockchain Ecosystem?
- How to Share Files in IPFS Blockchain?
- What is the CAP Theorem in blockchain

Interview Questions

- Top 50 Blockchain Interview Questions and Answers
- 15 Commonly Asked Blockchain Interview Questions & Answers

What programming languages are used for Blockchain?

We can use several programming languages to make our Blockchain-based projects :

What programming languages are used for Blockchain?

- Solidity
- C#
- C++
- SQL
- Ruby
- Java
- JavaScript
- Python
- GoLang

What programming languages are used for Blockchain?

Solidity is the majority-used programming language for blockchain-based projects.

How Blockchain works step by step?

In general, many nodes within the same blockchain implementation are required to perform algorithms to evaluate, validate, and process the history of the blockchain block whenever a new blockchain transaction is added to the blockchain. The fresh block of blockchain transaction is accepted into the ledger and the new block containing data is added to the blockchain if the majority of nodes confirm the authenticity of the block's history and signature.

What is a Blockchain Platform?

A blockchain platform allows developers and users to create transactions in the existing blockchain

database. For example, Ethereum has a native cryptocurrency called Ether (ETH). But the Ethereum blockchain allows users to create smart contracts and programmable tokens used in initial coin offerings (ICOs), and non-fungible tokens (NFTs). These are built upon the Ethereum infrastructure and secure by Ethereum network algorithms and system

What Is Cryptocurrency?

Cryptocurrencies are often traded on decentralized exchanges and can also be used to purchase goods and services. It is a digital or virtual currency that uses cryptography to encrypt its transactions and regulate the creation of new nodes or blocks.

What is a Blockchain Wallet?

Blockchain wallets are like bank accounts in that they allow users to easily transfer money between their various accounts. But where banks rely on centralized systems or government rules and protocols to hold customers' money, blockchain wallets use decentralized technology to manage transactions. It is a digital wallet that allows us to store, send, and receive cryptocurrencies like Bitcoin, Ethereum, Dogecoin, and Tether.

What is a Blockchain Wallet?

Get 90% Course fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today. The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

What is a Blockchain Wallet?

- Blockchain
- Tutorials

Similar Reads

Similar Reads

- [Company](#)
- [About Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Similar Reads

- [Explore](#)
- [Job-A-Thon Hiring Challenge](#)
- [Hack-A-Thon](#)
- [GfG Weekly Contest](#)
- [Offline Classes \(Delhi/NCR\)](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Similar Reads

- [Languages](#)
- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)

- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering

- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics

- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation

- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]

- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course

- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Web3.0: The Next Era of the Internet

-
-
-

Web3.0: The Next Era of the Internet

Web3.0, commonly known as Web3, is the next era of the Internet. It aims to address the limitations of the previous versions of the Web. From Web 1.0 to Web 2.0, we have been consuming information in a way that we have very little control over. However, Web3.0 claims that it will provide information on users' preferences and in the desired format.

Web3.0: The Next Era of the Internet

Web3.0: The Next Era of the Internet

In this article, we are going to explore the features of Web3.0 and how it is going to affect internet users like us. But first, it is necessary to understand how the World Wide Web has evolved over the years.

Versions of Web

The World Wide Web, or simply "Web", refers to the collection of web pages and resources that are accessed using the Internet. It is a tool used to share information and connect with others. To understand Web3, we need to go through the history and evolution of the Web.

Versions of Web

There are three versions of the Web:

Versions of Web

- Web 1.0
- Web 2.0
- Web 3.0

The Web 1.0

Web 1.0 was introduced by Timothy John Berners-Lee in the late 1980s. It was the era of read-only web as users were not able to interact with the websites. They can only consume the information without providing any feedback and suggestions. The technologies that were used during Web 1.0 were the HTML, URL and HTTP. This era lasted roughly until 2004.

Applications of Web 1.0

- Hyperlinks: The system of hyperlinks was the primary method of navigation, allowing users to click through to different pages within and across websites.
- Static Websites: The majority of web pages were static with only textual and visual content. They were built by web developers to promote companies and their services.

- CDN:Web1.0 served as a content delivery network (CDN). It allows its users to get a specific collection of content that is near to them.

The Web 2.0

With the release of Web 2.0 in the 21st century, a read-write web was introduced where users could not only consume content but also create and share it. It enables a two-way flow of information. However, this led to the centralization of data in the hands of a few tech companies which raises concerns about data privacy and ownership. The technologies that shaped Web 2.0 were the HTML5, JavaScript, CSS3 and different frameworks and libraries of modern programming languages.

Applications of Web 2.0

- Social Media: Platforms like Facebook (now known as Meta), Instagram and WhatsApp are major players in social media networks. These platforms changed the way of sharing information. They allow users to connect and interact with each other.
- Ed Tech: It is the short form of education technology. Currently, there are many companies that work in this field. Some prominent examples are GeeksForGeeks and Wikipedia. They encourage collaborative learning between students through the Internet. They offer virtual classrooms, online articles and tutorials to the students.
- OTT: It is an abbreviation that stands for Over The Top. It refers to the platforms that deliver on-demand media services directly to the users over the Internet. Websites like YouTube and Netflix provide a platform for creators to share and monetize their content.

What is Web 3.0?

Web 3.0 is defined as a decentralized Web architecture that is based on blockchain, artificial intelligence and token-based economics. We can also call it as a read-write-own web as it will allow its users to interact without any intermediaries and provide control over their data.

What is Web 3.0?

This new era of the Internet will provide more a personalized and customized web surfing experience to the users. It has potential to change the way of managing information and monetization of Internet. However, it is still in its early stages and there is ongoing debate regarding its development and implementation.

Features of Web 3.0

Web 3.0 which is the next era of the Internet provides several features that differentiate it from previous versions of the Web. These features are as follows -

Features of Web 3.0

- Decentralized: In Web3.0, the user's data might be stored at distinct locations and data sharing will be based on peer-to-peer connection. Some research suggests that Web3 will ensure the ownership of data to be maintained by its users. Currently, tech giants like Meta and Google have an unspoken agreement with their users that their data will belong to Web2.0 platforms which leads to centralization of data.
- Permissionless: Since Web3.0 will run on a decentralized peer-to-peer network, anyone can use it without any authorization and interference from the governing body. The exchange of information will rely on only the sender and receiver.
- AI and ML: Applying Artificial Intelligence and Machine Learning algorithms will make computers smarter than ever before. They can be able to understand the meaning or context of user requirements and needs.
- Connectivity: The resources and services available on the Internet will be more interconnected than ever. The information will become more accessible from anywhere, to everyone and through any application or platform.

Drawbacks of Web 3.0

The features of Web 3.0 look promising at first glance, but there are some drawbacks that we need to consider -

Drawbacks of Web 3.0

- **Data and Privacy:**In Web 3.0, users will have more control over their data as data storage will shift from a centralized to a decentralized model. However, as a result, individual users may face difficulties in managing and protecting their data. Scammers can misuse the user's data in digital fraud and phishing attacks. Therefore, Web3 users must be aware of its potential risks and challenges.
- **Complexity:**We all are aware that Web 3.0 is based on the blockchain and artificial intelligence. Developing and deploying applications using these technologies requires advanced skills and expertise. According to some experts, decentralization will make technology more complex and rather than making the Internet more accessible and simpler, it will become out of reach for basic users.
- **Regulatory Challenges:**Without any intermediaries and regulatory bodies, spreading hate speeches and misinformation becomes easy. Additionally, blockchain is immutable which means it is nearly impossible to track and erase data that could be troublesome.
- **Scalability:**The biggest challenge for Web3 is scalability. Unlike Web 2.0, which is based on a centralized model, Web 3.0 is based on a decentralized model, which may make it difficult for users to adopt. Due to its complexity and the required investment, the adoption process is expected to be slow.
- **Energy Consumption:**You must have heard about Bitcoin which is based on blockchain. Some researchers claim that the e-waste generated by Bitcoin mining can be equivalent to the amount of e-waste produced by a country like Netherlands. Moreover, running and saving data on a decentralized network consumes a tremendous amount of energy.

Applications of Web 3.0

Web 3.0 is going to shape a new digital environment which can be explained by the following applications -

Applications of Web 3.0

- Internet of Things: As the name suggests, IoT (A short form of Internet of Things) is a network of physical things. These physical objects could be anything from your kitchen appliances to cars and many other real-world objects. They are equipped with sensors and software that enable them to exchange data with other devices, such as your smartphone, over the internet.
- Decentralized Finance: Decentralized finance, or DeFi in short, is a modern financial technology that uses blockchain to manage financial transactions. Unlike Web 2.0 which requires a third-party (banks) intervention to complete a financial transaction, Web 3.0 eliminates the middleman by allowing us to perform transactions by using peer-to-peer financial networks.
- Non-Fungible Tokens: The non-fungible token, also known as NFTs, is a form of unique digital asset that can be bought and sold using cryptocurrencies like Bitcoin. This asset could be anything from the real world, say paintings, music, videos and so forth. Each NFT contains the information of seller, buyer and who owns it currently which makes it easily verifiable.

Trends and Technologies Driving Web 3.0

The vision of Web 3.0 is to create a Web that is less controlled by large companies through decentralized social networks. Despite of major developments in this new era of Internet, the adoption of Web 3.0 is still not accelerated. However, it is continuing.

Trends and Technologies Driving Web 3.0

Here is the list of trends and technologies driving Web 3.0 ?

Trends and Technologies Driving Web 3.0

- Use of Decentralized apps: Decentralized apps, also called as dApps, are those applications that are operated outside the authority of a single organization. Some of the examples of dApps are OpenSea, Uniswap, BitTorrent, etc.
- Metaverse: It is a three-dimensional virtual reality space where humans, as avatars, interact with each other and the environment. In October 2021, Facebook rebranded itself as Meta with a vision of building metaverse. The company is developing a VR social platform that allows users to enter a

computer-generated world and interact with it in a way that feels real.

- Development of Digital Currencies: In recent times, various decentralized organizations have launched digital currencies. They are only accessible through smartphones or computers over the Internet. Bitcoin and Ethereum are examples of such digital currencies.
- Smart Contracts: They are a type of transaction protocol that is automatically executed when the terms agreed by the buyer and seller are met. Once executed, the transactions can't be reversed but they are traceable and transparent.

Is Web 3.0 the Same as the Semantic Web?

In general, the word semantic refers to the study of meaning in language. A single word can have multiple meanings and people perceive it differently depending on the scenario. In the same way, search engines understand the meaning or context of a user's query and provide results based on their understanding.

Is Web 3.0 the Same as the Semantic Web?

Web 3.0 will be able to analyze all types of data on the internet, including media, emojis, and even the emotions behind search queries. This will lead to more personalized search results. Therefore, we can say that Web 3.0 is the same as the Semantic Web. According to Tim Berners Lee, the objective of the Semantic Web is to transform the data on the internet so that it can be easily interpreted by machines.

Conclusion

We are slowly making progress in the new era of Internet. It seems like the future of Internet will be Web 3.0. Hence, it is crucial to understand its potential and prepare for the changes it will bring. It promises a more interconnected and personalized internet, but it is not without its challenges. We need to make ourselves aware of the safety measures to protect ourselves from digital fraud and phishing attacks.

Conclusion

Become a Full-Stack Developer and also get 90% fee refund on completing 90% course in 90 days! Take the Three 90 Challenge today.

Conclusion

After successfully processing refunds worth over INR 5 Cr, GeeksforGeeks is back with the Three 90 challenge and this is your chance to upskill and get 90% refund. What more motivation do you need? Start the challenge right away!

Conclusion

S

Conclusion

- Web Technologies
- GBlog
- Web-Tech Blogs

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems

- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Types of Automation Testing

- Acceptance Testing - Software Testing
- Performance Testing - Software Testing
- Unit Testing - Software Testing
- Smoke Testing - Software Testing
- Non-Functional Testing
- Regression Testing - Software Engineering
- Keyword Driven Testing in Software Testing
- Functional Testing - Software Testing
- Integration Testing - Software Engineering
- Exploratory Testing

Automation Tools

- Selenium Automation tool - Software Engineering
- Selenium - Components, Features, Uses and Limitations
- Introduction to Cypress Testing Framework
- Basic Commands in Cypress
- What is Appium?
- Features of Appium
- Selenium vs Cypress - Which Framework is better to learn Selenium or Cypress?
- Difference between Cucumber Testing and Selenium Testing

Selenium Queries

- Selenium Locating Strategies

- Selenium WebDriver Commands
- Selenium Wait Commands - Implicit, Explicit, and Fluent Wait
- Components of Selenium
- Locating Strategies By XPath Using Java
- How to Handle Alert in Selenium using Java?
- How does Selenium perform mouse hover over an element?
- Exceptions - Selenium Python
- How does Selenium Webdriver Handle Proxy Authentication?
- How to capture Screen Shot in Selenium WebDriver?
- JavaScriptExecutor in Selenium
- Testing REST API with Postman and curl

Selenium ? Components, Features, Uses and Limitations

-
-
-

Selenium ? Components, Features, Uses and Limitations

Selenium is a powerful open-source framework for automating web browser testing easily. This article covers the basics of Selenium with including its components, features, uses, and limitations while providing a detailed view of it.

Selenium ? Components, Features, Uses and Limitations

Selenium is a powerful tool for controlling web browsers through programs. It is functional for all browsers, works on all major OS, and its scripts are written in various languages i.e., Python, Java, C#, etc., we will be working with Python. Selenium has four major components Selenium IDE, Selenium RC, Selenium Web driver, and Selenium GRID.

What is Selenium?

Selenium is a widely used tool for testing web-based applications that checks if they are doing as expected. It is a prominent preference amongst testers for cross-browser testing and is viewed as one of the most reliable systems for web application automation evaluation. Selenium is also platform-independent, so it can provide distributed testing using the Selenium Network. Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS and its scripts are written in various languages.

What is Selenium?

Selenium is like a building block in the journey of every software tester. Every tester starts his journey by learning Selenium. So if you want to learn Selenium to an advanced level and also want to learn other trending testing tools then you can check out our software testing course.

What is Selenium?

Selenium Features

Components

Selenium has been in the industry for a long time and is used by automation testers all around the globe.

Components

Let's check the four major components of Selenium

Components

Components of Selenium

1. Selenium IDE

Selenium IDE serves as an innovative toolkit for web testing, allowing users to record interactions with web applications. Selenium-IDE was initially created by Shinya Kasatani in 2006. Selenium IDE also helps to simplify the testing process. It is a friendly space for testers and developers to

team up. This helps everyone quickly share important testing information and results, making things work better and feel accomplished.

1. Selenium IDE

Features of Selenium are as follows:

1. Selenium IDE

- Record:With Selenium IDE, users can record how they use a web application.
- Playback:Selenium IDE automatically repeats what you recorded earlier.
- Browser Check:Selenium IDE works on various browsers for testing.
- Check Elements:Users can easily look at different parts of a webpage and set up how to work with them.
- Spotting Errors:Selenium IDE helps users find and fix issues in their automated tests, one step at a time.
- Exporting Tests:You can save tests created in Selenium IDE in different programming languages (like Java, Python, or C#). This lets you use them with other Selenium tools.

2. Selenium RC (Remote control)

Selenium Remote Control (RC)was one of the earliestSelenium tools, precedingWebDriver. It allowed testers to write automated web application tests in various programming languages like Java, C#, Python, etc. The key feature of Selenium RC was its ability to interact with web browsers using a server, which acted as an intermediary between the testing code and the browser.

2. Selenium RC (Remote control)

WebDriver is often considered the better choice over Selenium RC for several reasons are follows:

2. Selenium RC (Remote control)

- Improved API:WebDriver offers a more straightforward and intuitiveAPIcompared to Selenium RC, making it easier for developers and testers to write and maintain automated tests.

- **Better Performance:**WebDriver interacts directly with the browser, bypassing the need for an intermediary server like Selenium RC, which leads to faster test execution and improved performance.
- **Support for Modern Web Technologies:**WebDriver has better support for modern web technologies such as HTML5, CSS3, and JavaScript frameworks, ensuring compatibility with the latest web applications.

3. Selenium Web Driver

Selenium WebDriver is a robust open-source framework for automating web browsers, primarily aimed at easing the testing and verification of web applications. As an important part of the Selenium suite, WebDriver offers a programming interface to interact with web browsers, allowing developers and testers to automate browser actions seamlessly.

3. Selenium Web Driver

Features of Selenium Web Driver are as follows:

3. Selenium Web Driver

- **Direct Communication with Browsers:**Unlike Selenium RC, WebDriver interacts directly with the browser's native support for automation, leading to more stable and reliable testing.
- **Support for Parallel Execution:**WebDriver allows for parallel test execution, enabling faster test cycles and efficient utilization of resources.
- **Rich Set of APIs:**WebDriver provides a comprehensive set of APIs for navigating through web pages, interacting with web elements, managing windows, handling alerts, and etc.

4. Selenium GRID

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances.

4. Selenium GRID

Features of Selenium GRID are as follows:

4. Selenium GRID

- Selenium Grid allows running tests in parallel on multiple machines and managing different browser versions.
- The ability to run tests on remote browser instances is useful to spread the load of testing across several machines.
- Run tests in browsers running on different platforms or operating systems.

Features

The main Features of Selenium are as follows

Features

- Multi-Browser Support.
- Multi-Language Compatibility support.
- Easy Identification and Use of Web Elements while using the selenium.
- Performance and Speed are more as compared to the other tools.
- Dynamic Web Elements are present for easy use.
- Open Source platform for use.
- Portability (Easily accessible in any OS)
- Reusability of any codes and element is possible in the tool.

Features

For more, check out ?Features of Selenium Webdriver

Applications

Selenium is an open-source tool for automating web browsers. It helps to test the website functionality easily and check the regular performance across different browsers and systems

through cross-browser testing.

Applications

- Automated Web Application Testing: Selenium is mainly used for the automation testing of web applications across different browsers and platforms.
- Cross-Browser Testing: It checks the compatibility of web applications across various web browsers like Chrome, Firefox, Safari, and Edge.
- Web Scraping: Automates the filter of the data from websites for purposes such as data analysis and monitoring with easy automation.
- Continuous Integration/Continuous Deployment (CI/CD): Integrates with CI/CD tools like Jenkins to enable continuous testing as part of the development pipeline and automation.
- Functional Testing: It checks the functionality of a web application against the specified requirements of the browser.

Applications

For more, check ?Applications and Uses of Selenium WebDriver

Limitations

With concerning all these advantages of Selenium include some Limitations which are as follows:

Limitations

- Cross-Browser Compatibility: Selenium can give regular best results across multiple browsers, but sometimes it's restricted in that the web browsers understand and use the HTML and CSS differently from the respective browsers.
- Slow Test Execution: Because the automation depends on the various drivers with the browser that causes the process to slow. Selenium will be slow to respond when running tests on big web applications or websites.
- Difficulty in Handling Dynamic Web Elements: Selenium has difficulties in interacting with dynamic web elements like ID that will change on a web page sometime which causes the test script failure.

while testing the same.

- Limited Support for Mobile Applications: Selenium will not provide automation on mobile application testing, so developers choose the other tools or frameworks for automation purposes.
- Limited Support for Windows-based Applications: Developers will have to depend on third-party tools or libraries for Automation testing in desktop apps using Selenium.

Limitations

For more, check out: [Limitation of selenium](#).

Conclusion

Selenium tool will help test websites but it can be slow and cause problems with some things like changing a webpage. It is good to know its limits and use other tools when needed.

Which Selenium component is best?

Selenium WebDriver

What are the 4 parameters of Selenium?

URL, host, browser and port number.

Which XPath is best in Selenium?

Relative XPath.

Which XPath is best in Selenium?

Want to learn Software Testing and Automation to help give a kickstart to your career? Any student or professional looking to excel in Quality Assurance should enroll in our course, [Complete Guide to Software Testing and Automation](#), only on [GeeksforGeeks](#). Get hands-on learning experience with the latest testing methodologies, automation tools, and industry best practices through practical projects and real-life scenarios. Whether you are a beginner or just looking to build on existing skills, this course will give you the competence necessary to ensure the quality and reliability of software products. Ready to be a Pro in Software Testing? Enroll now and Take Your Career to a Whole New

Level!

Which XPath is best in Selenium?

N

Which XPath is best in Selenium?

- Selenium
- Software Testing
- Python-selenium

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)

- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects

- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design

- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Web Scraping

- Introduction to Web Scraping
- What is Web Scraping and How to Use It?
- Web Scraping - Legal or Illegal?
- Difference between Web Scraping and Web Crawling
- Web Scraping using cURL in PHP

Basics of Web Scraping

- HTML Basics
- Tags vs Elements vs Attributes in HTML
- CSS Introduction
- CSS Syntax
- JavaScript Cheat Sheet - A Basic Guide to JavaScript

Setting Up the Environment

- BeautifulSoup Installation - Python
- How to Install Requests in Python - For Windows, Linux, Mac
- Selenium Python Introduction and Installation
- How to Install Python Scrapy on Windows?

Extracting Data from Web Pages

- Implementing Web Scraping in Python with BeautifulSoup
- How to extract paragraph from a website and save it as a text file?
- Extract all the URLs from the webpage Using Python
- How to Scrape Nested Tags using BeautifulSoup?
- Extract all the URLs that are nested within tags using BeautifulSoup
- Clean Web Scraping Data Using clean-text in Python

Fetching Web Pages

- GET and POST Requests Using Python

- BeautifulSoup - Scraping Paragraphs from HTML

HTTP Request Methods

- GET method - Python requests
- POST method - Python requests
- PUT method - Python requests
- DELETE method- Python requests
- HEAD method - Python requests
- PATCH method - Python requests

Searching and Extract for specific tags BeautifulSoup

- Python BeautifulSoup - find all class
- BeautifulSoup - Search by text inside a tag
- Scrape Google Search Results using Python BeautifulSoup
- Get tag name using BeautifulSoup in Python
- Extracting an attribute value with beautifulsoup in Python
- BeautifulSoup - Modifying the tree
- Find the text of the given tag using BeautifulSoup
- Remove spaces from a string in Python
- Understanding Character Encoding
- XML parsing in Python
- Python - XML to JSON

Scrapy Basics

- Scrapy - Command Line Tools
- Scrapy - Item Loaders
- Scrapy - Item Pipeline
- Scrapy - Selectors

- Scrapy - Shell
- Scrapy - Spiders
- Scrapy - Feed exports
- Scrapy - Link Extractors
- Scrapy - Settings
- Scrapy - Sending an E-mail
- Scrapy - Exceptions

Selenium Python Basics

- Navigating links using get method - Selenium Python
- Interacting with Webpage - Selenium Python
- Locating single elements in Selenium Python
- Locating multiple elements in Selenium Python
- Locator Strategies - Selenium Python
- Writing Tests using Selenium Python

Python Web Scraping Tutorial

-
-
-

Python Web Scraping Tutorial

In today's digital world, data is the key to unlocking valuable insights, and much of this data is available on the web. But how do you gather large amounts of data from websites efficiently? That's where Python web scraping comes in. Web scraping, the process of extracting data from websites, has emerged as a powerful technique to gather information from the vast expanse of the internet.

Python Web Scraping Tutorial

In this tutorial, we'll explore various Python libraries and modules commonly used for web scraping and delve into why Python 3 is the preferred choice for this task. Along with this you will also explore how to use powerful tools like BeautifulSoup, Scrapy, and Selenium to scrape any website.

Essential Packages and Tools for Python Web Scraping

The latest version of Python, offers a rich set of tools and libraries specifically designed for web scraping, making it easier than ever to retrieve data from the web efficiently and effectively.

Essential Packages and Tools for Python Web Scraping

Table of Content

Essential Packages and Tools for Python Web Scraping

- Requests Module
- BeautifulSoup Library
- Selenium
- Lxml
- Urllib Module
- PyautoGUI
- Schedule
- Why Python3 for Web Scraping?

Requests Module

The requests library is used for making HTTP requests to a specific URL and returns the response. Python requests provide inbuilt functionalities for managing both the request and response.

Example: Making a Request

Python requests module has several built-in methods to make HTTP requests to specified URI using GET, POST, PUT, PATCH, or HEAD requests. A HTTP request is meant to either retrieve data from a specified URI or to push data to a server. It works as a request-response protocol between a client

and a server. Here we will be using the GET request. The GET method is used to retrieve information from the given server using a given URI. The GET method sends the encoded user information appended to the page request.

Example: Making a Request

Output

Example: Making a Request

Example: Making a Request

For more information, refer to our [Python Requests Tutorial](#).

BeautifulSoup Library

Beautiful Soup provides a few simple methods and Pythonic phrases for guiding, searching, and changing a parse tree: a toolkit for studying a document and removing what you need. It doesn't take much code to document an application.

BeautifulSoup Library

Beautiful Soup automatically converts incoming records to Unicode and outgoing forms to UTF-8. You don't have to think about encodings unless the document doesn't define an encoding, and BeautifulSoup can't catch one. Then you just have to choose the original encoding. BeautifulSoup sits on top of famous Python parsers like LXML and HTML, allowing you to try different parsing strategies or trade speed for flexibility.

Example

- **Importing Libraries:** The code imports the requests library for making HTTP requests and the BeautifulSoup class from the bs4 library for parsing HTML.

- **Making a GET Request:** It sends a GET request to <https://www.geeksforgeeks.org/python-programming-language/> and stores the response in the variable r.

- Checking Status Code:It prints the status code of the response, typically 200 for success.
- Parsing the HTML: The HTML content of the response is parsed using BeautifulSoup and stored in the variable soup.
- Printing the Prettified HTML:It prints the prettified version of the parsed HTML content for readability and analysis.

Example

Output

Example

Finding Elements by Class

Now, we would like to extract some useful data from the HTML content. The soup object contains all the data in the nested structure which could be programmatically extracted. The website we want to scrape contains a lot of text so now let?s scrape all those content. First, let?s inspect the webpage we want to scrape.

Finding Elements by Class

Finding Elements by Class

Finding Elements by Class

In the above image, we can see that all the content of the page is under the div with class entry-content. We will use the find class. This class will find the given tag with the given attribute. In our case, it will find all the div having class as entry-content.

Finding Elements by Class

We can see that the content of the page is under the <p> tag. Now we have to find all the p tags present in this class. We can use thefind_allclass of the BeautifulSoup.

Finding Elements by Class

Output:

Finding Elements by Class

Finding Elements by Class

For more information, refer to our [Python BeautifulSoup](#).

Selenium

Selenium is a popular Python module used for automating web browsers. It allows developers to control web browsers programmatically, enabling tasks such as web scraping, automated testing, and web application interaction. Selenium supports various web browsers, including Chrome, Firefox, Safari, and Edge, making it a versatile tool for browser automation.

Example 1: For Firefox

In this specific example, we're directing the browser to the Google search page with the query parameter `?geeksforgeeks?`. The browser will load this page, and we can then proceed to interact with it programmatically using Selenium. This interaction could involve tasks like extracting search results, clicking on links, or scraping specific content from the page.

Example 1: For Firefox

Output

Example 1: For Firefox

Example 2: For Chrome

- We import the webdriver module from the Selenium library.
- We specify the path to the web driver executable. You need to download the appropriate driver for your browser and provide the path to it. In this example, we're using the Chrome driver.
- We create a new instance of the web browser using `webdriver.Chrome()` and pass the path to the Chrome driver executable as an argument.

- We navigate to a webpage by calling the `get()` method on the browser object and passing the URL of the webpage.
- We extract information from the webpage using various methods provided by Selenium. In this example, we retrieve the page title using the `title` attribute of the browser object.
- Finally, we close the browser using the `quit()` method.

Example 2: For Chrome

Output

Example 2: For Chrome

Example 2: For Chrome

For more information, refer to our [Python Selenium](#).

Lxml

The `lxml` module in Python is a powerful library for processing XML and HTML documents. It provides a high-performance XML and HTML parsing capabilities along with a simple and Pythonic API. `lxml` is widely used in Python web scraping due to its speed, flexibility, and ease of use.

Example

Here's a simple example demonstrating how to use the `lxml` module for Python web scraping:

Example

- We import the `html` module from `lxml` along with the `requests` module for sending HTTP requests.
- We define the URL of the website we want to scrape.
- We send an HTTP GET request to the website using the `requests.get()` function and retrieve the HTML content of the page.
- We parse the HTML content using the `html.fromstring()` function from `lxml`, which returns an HTML element tree.
- We use XPath expressions to extract specific elements from the HTML tree. In this case, we're

extracting the text content of all the <a> (anchor) elements on the page.

- We iterate over the extracted link titles and print them out.

Example

Output

Urllib Module

The urllib module in Python is a built-in library that provides functions for working with URLs. It allows you to interact with web pages by fetching URLs (Uniform Resource Locators), opening and reading data from them, and performing other URL-related tasks like encoding and parsing. Urllib is a package that collects several modules for working with URLs, such as:

Urllib Module

- urllib.request for opening and reading.
- urllib.parse for parsing URLs
- urllib.error for the exceptions raised
- urllib.robotparser for parsing robot.txt files

Urllib Module

If urllib is not present in your environment, execute the below code to install it.

Example

Here's a simple example demonstrating how to use the urllib module to fetch the content of a web page:

Example

- We define the URL of the web page we want to fetch.
- We use urllib.request.urlopen() function to open the URL and obtain a response object.
- We read the content of the response object using the read() method.
- Since the content is returned as bytes, we decode it to a string using the decode() method with

?utf-8? encoding.

- Finally, we print the HTML content of the web page.

Example

Output

Example

PyautoGUI

The pyautogui module in Python is a cross-platform GUI automation library that enables developers to control the mouse and keyboard to automate tasks. While it's not specifically designed for web scraping, it can be used in conjunction with other web scraping libraries like Selenium to interact with web pages that require user input or simulate human actions.

Example

In this example, pyautogui is used to perform scrolling and take a screenshot of the search results page obtained by typing a query into the search input field and clicking the search button using Selenium.

Example

Output

Example

Schedule

The schedule module in Python is a simple library that allows you to schedule Python functions to run at specified intervals. It's particularly useful in web scraping in Python when you need to regularly scrape data from a website at predefined intervals, such as hourly, daily, or weekly.

Example

- We import the necessary modules: schedule, time, requests, and BeautifulSoup from the bs4

package.

- We define a function `scrape_data()` that performs the web scraping task. Inside this function, we send a GET request to a website (replace `?https://example.com?` with the URL of the website you want to scrape), parse the HTML content using BeautifulSoup, extract the desired data, and print it.
- We schedule the `scrape_data()` function to run every hour using `schedule.every().hour.do(scrape_data)`.
- We enter a main loop that continuously checks for pending scheduled tasks using `schedule.run_pending()` and sleeps for 1 second between iterations to prevent the loop from consuming too much CPU.

Example

Output

Example

Why Python3 for Web Scraping?

Python's popularity for web scraping stems from several factors:

Why Python3 for Web Scraping?

Ease of Use: Python's clean and readable syntax makes it easy to understand and write code, even for beginners. This simplicity accelerates the development process and reduces the learning curve for web scraping tasks.

Why Python3 for Web Scraping?

Rich Ecosystem: Python boasts a vast ecosystem of libraries and frameworks tailored for web scraping. Libraries like BeautifulSoup, Scrapy, and Requests simplify the process of parsing HTML, making data extraction a breeze.

Why Python3 for Web Scraping?

Versatility: Python is a versatile language that can be used for a wide range of tasks beyond web

scraping. Its flexibility allows developers to integrate web scraping seamlessly into larger projects, such as data analysis, machine learning, or web development.

Why Python3 for Web Scraping?

Community Support: Python has a large and active community of developers who contribute to its libraries and provide support through forums, tutorials, and documentation. This wealth of resources ensures that developers have access to assistance and guidance when tackling web scraping challenges.

Conclusion

this tutorial has shown you the basics of how to use Python for web scraping. With the tools we've discussed, you can start collecting data from the internet quickly and easily. Whether you need this data for a project, research, or just for fun, Python makes it possible. Remember to always scrape data responsibly and follow the rules set by websites. If you're excited to learn more about Python and web scraping, check out our [Python Course](#). It's a great resource to deepen your understanding and enhance your skills, all while having fun exploring the power of Python.

Conclusion

Python Web Scraping ? FAQs

1.What is Python web scraping?

Python web scraping refers to the process of extracting data from websites using Python programming. It involves fetching HTML content from a web page and parsing it to gather specific information.

2.Is web scraping legal?

Web scraping is legal as long as you comply with the website's terms of service and avoid scraping personal or sensitive data. Always check the site's robots.txt file to ensure you're following the rules.

3.What is the difference between BeautifulSoup and Scrapy?

BeautifulSoup is a simpler library for beginners focused on HTML parsing and extraction, whereas Scrapy is a more advanced web scraping framework that can handle complex tasks like crawling large datasets or handling pagination automatically.

4.What are some common use cases for Python web scraping?

Common use cases include extracting data for price comparison, content aggregation, job listings, real estate data, and sentiment analysis. Web scraping helps gather structured data from websites for various business and research purposes.

4.What are some common use cases for Python web scraping?

4.What are some common use cases for Python web scraping?

4.What are some common use cases for Python web scraping?

Level up your coding with DSA Python in 90 days! Master key algorithms, solve complex problems, and prepare for top tech interviews. Join the Three 90 Challenge? complete 90% of the course in 90 days and earn a 90% refund. Start your Python DSA journey today!

4.What are some common use cases for Python web scraping?

- AI-ML-DS
- Python
- Web-scraping

4.What are some common use cases for Python web scraping?

- python

Similar Reads

Similar Reads

- Company
- About Us

- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL

- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML

- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science

- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies

- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor

- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

Introduction to Ethical Hacking

- What is Hacktivism ?
- Methodology followed by the Hackers
- Remote Access in Ethical Hacking
- Kali Linux - Information Gathering Tools
- ARIN in Ethical Hacking
- Basic Characteristics of Computer Networks

Foot Printing and Reconnaissance

- What is DNS Footprinting?
- Footprinting Through Search Engines
- What is Whois Footprinting?
- Footprinting Using Social Engineering Method

Scanning Networks

- What is Credentialed Vulnerability Scan?
- What are Scanning Attacks?
- Malware Scan in Ethical Hacking
- What is Running of a Malware Scan?
- WAScan - web application security scanner in Kali Linux
- What is TCP-ACK Scanning?
- Port Scanning Techniques By Using Nmap
- What is SYN Scanning?
- What is UDP Scanning?

Enumeration

- Cyber Security - Types of Enumeration
- What is DNS Enumeration?
- SMTP Enumeration
- LDAP Enumeration
- What is NTP Enumeration?
- What is IPsec Enumeration?
- What is NetBIOS Enumeration?
- SNMP Enumeration
- What is Security Testing in Enumeration?

System Hacking

- What is System Hacking in Ethical Hacking?
- What is Windows Hacking ?
- Importance of Physical Security in Ethical Hacking
- What is Non-Electronic Password Attack on a System?
- What is Password Guessing Attack?
- Credential Stuffing in Ethical Hacking

- Reverse Brute Force Attack in System Hacking
- Brute Force Attack
- What is a Default Password Attack Threat?
- USB Drop Attack in System Hacking
- What is Sniffing Attack in System Hacking?
- How to Prevent Man In the Middle Attack?
- How To Generate Rainbow Table Using WinRTGen?
- What is Elcomsoft Distributed Password Recovery?
- pwdump7 in System Hacking
- FGDUMP in System Hacking
- Password Auditing With L0phtcrack 7 Tool
- What is Salted Password Hashing?
- How to Hack Wifi Using Aircrack-ng in Termux Without Root?
- How to Defend Against Password Cracking of Systems?
- How to Defend Against Wi-Fi Pineapple?
- What is DLL Hijacking?
- How to Prevent Privilege Escalation?

Malware Analysis

- Most Popular Methods Used By Hackers to Spread Ransomware
- What is Malvertising?
- How to Find Trojan on Computers?
- Malwares - Malicious Software
- What is WannaCry? How does WannaCry ransomware work?
- Working of Stuxnet Virus
- CryptoLocker Ransomware Attack
- Storm Worm

- What is Zeus Malware?
- What is SQL Slammer Virus?
- How to Install Trojan Virus on Any Computer?
- Different Ways to Remove Trojan Horse Malware
- How to Defend Against Botnets ?
- What is Proxy Trojan?
- What are Banking Trojans?
- What is a Computer Virus?
- Virus Hoax
- Difference between Worms and Virus
- Port Scanning Attack
- What is System Integrity Check?
- Code Emulation Technique For Computer Virus Detection
- Heuristic Virus
- How to Prevent Backdoor Attacks?

Sniffing

- What are Active Sniffing Attacks?
- What is Protocol Analyzer?
- What is MAC Spoofing Attack?
- How to Prevent MAC Flooding?
- What is Port Stealing?
- Dynamic Host Configuration Protocol (DHCP)
- DHCP Starvation Attack
- What is Rogue DHCP Server Attack?
- What is ARP Spoofing Attack?
- How to Prevent DNS Poisoning and Spoofing?

- DNS Spoofing or DNS Cache poisoning
- How to Detect Sniffer in Your Network?
- Mitigation of DHCP Starvation Attack

Social Engineering

- Social Engineering - The Art of Virtual Exploitation
- What is Insider Attack?
- What is an Impersonation Attack?
- What are Tailgating?
- How Hackers Use Social Engineering to Get Passwords on Facebook?
- Pretexting in Social Engineering
- Credit Card Frauds
- Active Social Engineering Defense (ASED)
- Cyber Crime - Identity Theft
- Penetration Testing - Software Engineering

Denial-of-Service

- Distributed Denial of Service DDoS attack
- What are Bandwidth Attacks?
- HTTP Flood Attack
- ICMP Flood DDoS Attack
- Ping Flood Attack
- What is a Permanent DoS (PDoS) Attack?
- What is Phlashing?

Session Hijacking

- TCP/IP Hijacking
- UDP Session Hijacking

- What are Types of Session Hijacking ?
- Difference Between Spoofing and Hijacking
- Application Level Hijacking Using Proxy Hacking
- Man-in-the-Browser Attack
- DOM-Based Cookie Manipulation
- What are Session Replay Attacks?
- What is Cookie Hijacking?
- Session Prediction Software Attack
- Session Hijacking
- Types of Client-Side Attacks
- Difference Between XSS and SQL Injection
- How SYN cookies are used to preventing SYN Flood attack
- IPSec Architecture

Evading IDS,Firewalls,and Honeypots

- Bypass Firewalls Using SSH
- What is Bitvise SSH Client?
- How to Prevent Port Scan Attacks?
- What is Source Port Randomization For Caching DNS ?
- Types of Evasion Technique For IDS

Hacking Web Servers

- Web Threat Shield
- Web Reputation
- What is Recursive DNS?
- Path Traversal Attack and Prevention
- What is Server Misconfiguration?
- Web Cache Poisoning

- How to Brute-Force SSH in Kali Linux?
- How to Hack a Web Server?
- Testing For IMAP SMTP Injection
- Web Parameter Tampering Attack on Web Servers
- How To Crack Online Web Form Passwords?
- How to Crack FTP Passwords?
- Cookie Tampering Techniques
- What is Input Validation Attack?
- Ethical Hacking | Footprinting
- Parsero - Tool for reading the Robots.txt file in Kali Linux
- What is Credential Harvester Attack ?
- Script Http-Userdir-Enumeration Method
- Linux - Metasploit Command
- Working with Payload Metasploit in Kali Linux
- What is Code Access Security?
- CRLF Injection Attack
- What is Patch Management?

Hacking Web Applications

- What is Cookie Poisoning?
- What are Injection Flaws?
- How to Prevent Broken Access Control?
- What is Improper Error Handling?
- What is Log Tampering?
- OWASP Top 10 Vulnerabilities And Preventions
- Insecure Cryptographic Storage Vulnerability
- Web Server and its Types of Attacks

- Insufficient Transport Layer Protection
- What is Failure to Restrict URL Access?
- Session Fixation Attack
- What is Malicious File Execution?
- What is CSV Injection?
- Converting a CVE list to Patch Vulnerabilities
- What is Arbitrary Code Execution?
- Malicious Script
- What is User Privileges in Ethical Hacking ?
- What is CAPTCHA Attack?
- What is Banner Grabbing?
- WhatWaf - Detect And Bypass Web Application Firewalls And Protection Systems
- User Directed Spidering with Burp
- What is Attack Surface?
- What is Authentication Attack?
- User Enumeration in Ethical Hacking
- What is SMTP Header Injection?
- What is Canonicalization Attack?
- How a Connection String Injection Attack is Performed?
- What is Connection String Parameter Pollution?
- Pass-the-Hash (PtH) Attack
- What is WSDL Attack?

Hacking Wireless Networks

- Orthogonal Frequency-Division Multiplexing (OFDM)
- Direct Sequence Spread Spectrum in Wireless Networks
- Frequency-Hopping Spread Spectrum in Wireless Networks

- Warchalking in Wireless Networks
- Types of WiFi Antenna in Wireless Networks
- Types of Wireless Security Encryption
- WEP Crack Method in Wireless Networks
- Bluesnarfing Attack in Wireless Networks
- BlueSmack Attack in Wireless Networks
- How To Install Super Bluetooth Hack on Android?
- Bluebugging in Wireless Networks

Cloud Computing

- Net-Centric Computing in Cloud Computing
- Security Issues in Cloud Computing
- Packet Switched Network (PSN) in Networking
- What is Parallel File System in Cloud Computing?
- How To Install AWS CLI - Amazon Simple Notification Service (SNS)?
- How to Authorize Inbound Traffic For Your Linux Instances?
- How To Multiple IP Addresses Work in Ethical Hacking?

Cryptography

- What is Heartbleed Bug in Ethical Hacking ?
- Email Hijacking
- What is Hybrid Cryptosystem in Ethical Hacking?

Ethical Hacking Tutorial

-
-
-

Ethical Hacking Tutorial

This Ethical Hacking tutorial covers both basic and advanced concepts of Ethical Hacking. Whether you are a beginner or an experienced cybersecurity professional, this tutorial is the perfect resource to learn how to tackle vulnerabilities and weaknesses in systems before malicious hackers can exploit them.

Ethical Hacking Tutorial

From understanding the hacking basics to diving into advanced ethical hacking techniques, including Footprinting and Reconnaissance, Scanning Networks, Enumeration, and more.

Ethical Hacking Tutorial

This ethical hacking tutorial covers everything you need to know to become a skilled ethical hacker and learn how to use your skills to find and fix vulnerabilities in systems before they can be exploited by malicious hackers. But before starting, this ethical hacking tutorial, let's understand what hacking is.

Ethical Hacking Tutorial

Ethical Hacking Tutorial

What is hacking?

In simple terms, hacking is the practice of exploiting vulnerabilities in computer systems, networks, or applications to gain unauthorized access. Hackers use techniques such as phishing, brute force attacks, and SQL injection to break into systems. They often employ tools like keyloggers, malware, and Trojan horses to steal sensitive data or disrupt services. Whether it's cracking passwords, bypassing firewalls, or using zero-day exploits, hacking involves manipulating systems to achieve their goals, often staying a step ahead of security measures.

What is Ethical Hacking?

Ethical Hacking, also referred to as "white hat hacking," "Pen Testing," or simply "ethical hacking," plays a critical role in maintaining the security and integrity of computer systems and networks. It involves cybersecurity practices that use hacking tools and techniques to identify

vulnerabilities and weaknesses in computer systems and networks with the primary objective of preventing unauthorized access to systems and sensitive data, protecting against cyber-attacks, and ensuring the security of an organization's assets.

Introduction to Ethical Hacking

- What is hacktivism?
- Scanning Methodology for ethical hacking?
- What is remote access to ethical hacking?
- Information gathering techniques for ethical hacking
- What is ARIN in ethical hacking?
- Characteristics of information on the computer.

Foot Printing and Reconnaissance

- What is DNS Footprinting
- Footprinting through search engines
- What is whois footprinting
- Footprinting through the Social Engineering method

Scanning Networks

- What is a credentialed vulnerability scan
- What are scanning attacks
- What are the types of scanning used in ethical hacking
- What is malware scan in ethical hacking
- What is Running a malware scan
- What is web application scanning
- What is Port Scan for ethical hacking
- What is TCP scanning?
- What is FIN scanning?

- What is SYN scanning?
- What is UDP scanning?
- What is TCP-ACK scanning?
- What is TCP Window Scan?
- What is IPv6 Scanning

Enumeration

- What is Enumeration
- What is DNS Enumeration?
- What is SMTP enumeration?
- What is LDAP Enumeration?
- What is NTP Enumeration?
- IPsec Enumeration
- What is RPC Enumeration?
- What is SMB Enumeration?
- What is NetBIOS Enumeration?
- What is SNMP Enumeration?
- What is Security Testing in Enumeration?

System Hacking

- What is system hacking?
- Windows hacking
- What is the importance of physical security in system hacking?
- What is a non-electronic password attack on a system?
- What is a password-guessing attack?
- What is credential stuffing in system hacking?
- What is a reverse brute force attack in system hacking?
- How to prevent brute force attacks on any system?

- How to Protect Your WordPress Site from Brute Force Attacks?
- What is the default password attack threat?
- What is a USB drop attack in system hacking?
- What is a sniffing attack on system hacking?
- How to prevent a man-in-the-middle attack?
- How to Generate Rainbow Table Using WinRTGen?
- What is Elcomsoft Distributed Password Recovery?
- What are NTLM Relay Attacks in system hacking?
- What is pwdump7?
- What is fgdump?
- Steps of Password Auditing with the L0phtcrack 7 tool
- How to Use Ophcrack for Windows Password Recovery?
- What is Salted Password Hashing?
- How to Crack WPA/WPA2 using Aircrack-Ng?
- How to defend against Password Cracking of systems?
- What is Wi-Fi Pineapple and How do you defend against it?
- How to keep your computer secure at home?
- What is DLL Hijacking?
- How to Prevent Privilege Escalation?

Malware Analysis

- The most popular methods used by hackers to spread ransomware
- What is Malvertising?
- How to find Trojans on computers?
- What is a Malicious code?
- What is a WannaCry ransomware attack?
- What is a Stuxnet ransomware attack?

- What is a CryptoLocker ransomware attack?
- What is Storm Worm?
- Some Details about Zeus (malware)
- What is Mydoom malware?
- What is SQL Slammer Virus?
- How to install a Trojan virus on any computer?
- Different Ways to Remove Trojan Horse Malware
- Different Types of Botnet Attacks
- What is Proxy Trojan?
- What Are Banking Trojans?
- Symptoms and Effects of Computer Viruses
- What is a virus hoax?
- What is a computer worm?
- What is the port scanning attack?
- Monitoring of Suspicious Network Activity
- What is a system integrity check?
- What is the Code Emulation Technique for Computer Virus Detection?
- What Is a Heuristic Virus and How to Remove It?
- What Is a Backdoor & How to Prevent Backdoor Attacks?

Sniffing

- What Are Active Sniffing Attacks?
- What is a Protocol analyzer?
- What is PRISM (surveillance program)?
- What is a MAC Spoofing Attack?
- What is MAC Flooding? How to prevent it?
- What is Port Stealing?

- methods of DHCP Request/Reply Messages
- What is a DHCP Starvation Attack?
- What is Rogue DHCP Server Attack?
- What Is ARP Spoofing Attack?
- How to Prevent DNS Poisoning and DNS Spoofing
- What is DNS cache poisoning?
- How to Detect Sniffer in Your Network?
- Mitigation of DHCP starvation attack.

Social Engineering

- Types of Social Engineering Attacks
- What are Malicious Apps?
- What is an insider attack?
- Insider Threat Prevention Best Practices
- What is an Impersonation Attack?
- What are intimidation threats and attacks?
- What is Tailgating?
- How Hackers Use Social Engineering to Get Passwords on Facebook
- Different Ways Social Media Impacts Your Risk Profile
- What is Pretexting
- What is credit card fraud?
- What are Social Engineering Countermeasures?
- How to Protect Yourself From Identity Theft?
- Penetration Testing
- What is Active Social Engineering Defense (ASED)

Denial-of-Service

- What are the Most Common Types of DDoS Attacks?

- What are Bandwidth Attacks?
- What is an HTTP flood attack?
- What is an SYN flood attack?
- What is an ICMP Flood DDoS Attack?
- What is a ping flood attack?
- What is a Permanent DoS (PDoS) Attack?
- What is phishing?
- Sabotage Attack Detection for Additive Manufacturing Systems
- What is Bricking a System?
- What are Application layer attacks?
- What is the Distributed Reflection Denial of Service?
- some DDoS Attack Tools
- What is Attack Mitigation?
- What is the TCP intercept feature?
- What is ingress filtering?
- What is egress filtering?
- Explain throttling DDoS attacks using a discrete logarithm problem
- How to Defend Against Botnets?
- What is black hole filtering?

Session Hijacking

- What is TCP/IP Hijacking?
- What is UDP hijacking?
- What are the Types of Session Hijacking?
- What is different between Spoofing and hijacking?
- What is Application Level Hijacking using proxy hacking?
- What is a Man-in-the-browser attack?

- What is DOM-based cookie manipulation?
- What Are Session Replay Attacks?
- What is Cookie Hijacking? (And How to Prevent It)
- What is a Session Prediction software attack?
- What is Qrljacking software attack in session hijacking?
- What is a Session fixation software attack in session hijacking?
- Types of client-side attacks
- What is the Difference Between XSS and SQL Injection
- What is a TCP SYN Flood attack?
- How To Prevent A SYN Flood Attack
- AH and ESP protocols in IPsec

Evading IDS, Firewalls, and Honeypots

- How to Bypass Firewalls Using SSH?
- What is Bitwise SSH Client?
- What Is a Port Scan? How to Prevent Port Scan Attacks?
- What is the Source Port Randomization for Caching DNS?
- Types of Evasion Techniques for IDS

Hacking Web Servers

- What is a web threat shield?
- What-is-web-reputation ?
- What is recursive DNS?
- What is Path Transversal Attack?
- What is Server Misconfiguration?
- What is Web cache poisoning?
- What is an SSH Brute force attack?
- How to Hack a Web Server?

- Testing for IMAP SMTP Injection on any web server
- What is a Web Parameter Tampering attack on web servers?
- How to Crack Online Web Form Passwords
- How to crack FTP passwords?
- What are COOKIE TAMPERING TECHNIQUES?
- What is an input validation attack?
- What is Website Footprinting?
- Methods of information gathering using robots.txt
- What Is a Credential Harvester Attack?
- What is the Script http-userdir-enumeration method?
- List of Metasploit Windows Exploits
- List of PAYLOADS IN METASPLOIT
- List of AUXILIARY MODULE REFERENCE in Metasploit
- Relevance of NOPs in Metasploit
- Ways to Protect Your Web Servers
- What is Code Access Security?
- What is HTTP Response Splitting?
- What is Patch Management?
- How to install patches of Red Hat Linux manually?

Hacking Web Applications

- What Is Cookie Poisoning?
- What are injection flaws?
- What is Broken Access Control?
- What is Improper Error Handling?
- What is Log tampering?
- What is a security misconfiguration?

- What is Insecure Cryptographic Storage vulnerability?
- What are Web Services Attacks?
- What is Insufficient Transport Layer Protection?
- What is a Protocol DDoS Attack?
- What is Failure to Restrict URL Access?
- What is a session fixation attack?
- What is malicious file execution?
- What is CSV Injection?
- Converting a CVE list to Patch vulnerabilities
- What is Arbitrary code execution?
- What is Malicious Script?
- What are user privileges in ethical hacking?
- What is the CAPTCHA attack?
- What is WHOIS LOOKUP?
- What is Banner Grabbing?
- How to Find Which WAF is Protecting a Website
- What is User Directed Spidering with Burp?
- What is Attack Surface?
- What is an Authentication Attack?
- What Is User Enumeration?
- What is SMTP header injection?
- What is a Canonicalization attack?
- How a connection string injection attack is performed?
- What is Connection String Parameter Pollution?
- What is a Pass-the-Hash (PTH) Attack?
- What is a WSDL attack?
- How to Protect Against SQL Injection Attacks

SQL Injection

- What are the Types of SQL Injection (SQLi)?
- Explain Error-Based SQL Injections
- What is SQL injection UNION attacks

Hacking Wireless Networks

- What is orthogonal frequency-division multiplexing (OFDM)?
- What is the Direct-sequence spread spectrum in Wireless Networks?
- What is the Frequency-hopping spread spectrum in Wireless Networks?
- What is War chalking in Wireless Networks?
- Different types of Wi-Fi Antenna in Wireless Networks?
- Different kinds of Wireless Security Encryption
- What is the Simple WEP Crack method in Wireless Networks?
- How to Hack Bluetooth Devices?
- What is Blue Smack Attack in Wireless Networks?
- What is a Blue snarfing attack in Wireless Networks?
- What is Blue Bugging in Wireless Networks?

Cloud Computing

- What is Net-Centric Computing in cloud computing?
- Cloud Vulnerabilities in cloud computing
- What is Packet Switched Network (PSN) in cloud computing?
- What is a parallel file system in cloud computing?
- How to Install AWS CLI ? Amazon Simple Notification Service (SNS)
- How to authorize inbound traffic for your Linux instances?
- How do multiple IP addresses work in ethical hacking?

Cryptography

- What does the Heart bleed Bug in ethical hacking?
- What is the One-Way Hash Function in ethical hacking?
- What is Email Security in ethical hacking?
- What is a Hybrid cryptosystem in ethical hacking?

Why is Ethical Hacking Important & Job Roles for Ethical Hackers?

Nowadays, Data is the most valuable asset for any corporation. In order to the preservation of privacy and Integrity of data from Malicious Attacks Ethical hacking is Important because it helps organizations identify and mitigate potential security risks, prevent cyber attacks, and improve overall security. Ethical hackers play a critical role in this process by identifying vulnerabilities, conducting penetration testing, and recommending solutions to improve security.

Why is Ethical Hacking Important & Job Roles for Ethical Hackers?

Roles of Ethical Hackers:-

Why is Ethical Hacking Important & Job Roles for Ethical Hackers?

- Identifies the Vulnerability of Computer Systems, networks and software applications.
- Simulates Cyber Attacks to Identify Weaknesses in Security Control and Systems.
- Ethical Hackers maintain documents of their findings, provide recommendations and Improve Overall Security.
- Ethical Hacker assists the organization to implement Security Policies and procedure to enhance overall Security.
- Ethical hackers also provide security training to other employees.

What are Security Threats and their types?

Any Activity that has the potential to harm a system and breach the security of the Organization.

What are Security Threats and their types?

Types of Security Threats:-

What are Security Threats and their types?

There are mainly two types of Security Threats:-

What are Security Threats and their types?

- Physical Threats? Physical threats are those that involve physical access to an organization's assets. Internal e.g. hardware fire, faulty power supply, internal hardware failures etc External e.g. floods, fires, earthquakes etc Humane e.g. vandalism, arson, accidental errors etc
- Internal e.g. hardware fire, faulty power supply, internal hardware failures etc
- External e.g. floods, fires, earthquakes etc
- Humane e.g. vandalism, arson, accidental errors etc
- Non-Physical Threats? Non-physical threats include every threat that has no physical manifestation. They are also known as logical threats. Here are some of the most common types of security threats: Malware Phishing Denial of Service (DoS) attacks Insider threats Advanced Persistent Threats (APTs) Zero-day attacks Man-in-the-middle attacks Password attacks
- Malware
- Phishing
- Denial of Service (DoS) attacks
- Insider threats
- Advanced Persistent Threats (APTs)
- Zero-day attacks
- Man-in-the-middle attacks
- Password attacks

What are Security Threats and their types?

- Internal e.g. hardware fire, faulty power supply, internal hardware failures etc
- External e.g. floods, fires, earthquakes etc
- Humane e.g. vandalism, arson, accidental errors etc

What are Security Threats and their types?

- Malware
- Phishing
- Denial of Service (DoS) attacks
- Insider threats
- Advanced Persistent Threats (APTs)
- Zero-day attacks
- Man-in-the-middle attacks
- Password attacks

Advantages of Ethical Hacking

Here are various advantages of hacking:-

Advantages of Ethical Hacking

- It is used to test how good security is on your network.
- It is used to recover the loss of information, especially when you lost your password.
- It is used to perform penetration testing to increase the security of the computer and network.

Disadvantages of Ethical Hacking

- Ethical hacking activities, if not managed properly, can expose systems to new vulnerabilities.
- Ethical hackers must operate within legal boundaries, and any misstep can lead to legal consequences.
- Requires a deep understanding of various technologies, making it difficult to find qualified professionals.
- Relying simple on ethical hacking can create a false sense of security; continuous monitoring is essential.
- Dependence on Ethical Hackers: Over-reliance on ethical hackers can neglect the importance of a security-aware culture.

- Risk that an ethical hacker might misuse their knowledge or access.

Disadvantages of Ethical Hacking

Conclusion

Disadvantages of Ethical Hacking

This Ethical Hacking tutorial has provided you with the foundational knowledge and tools to understand the importance and techniques of ethical hacking. By mastering these skills, you can help protect systems and data from malicious attacks, ensuring a safer digital environment. Ethical hacking not only strengthens security but also promotes a proactive approach to identifying and mitigating potential threats. As you continue to develop your expertise, remember that ethical hacking is a continuous learning journey, requiring regular updates and adaptation to new challenges.

Q1. What is Ethical Hacking?

Ethical hacking, also known as "White Hat" hacking, is the practice of using computer skills to test and secure computer systems and networks. Ethical hackers use their skills to find and fix vulnerabilities and weaknesses in systems before they can be exploited by malicious hackers, also known as "black hat" hackers.

Q2. Is an Ethical hacking course tough?

Ethical Hacking requires a high level of knowledge of operating systems, computer networking, and any programming language. Ethical hacking is not difficult to learn but requires great computer skills.

Q3. What is the Ethical Hacking certification cost?

For taking CEH v12 exam in India, you need to pay around ₹45,543 approx.

Q4. What is the salary of an ethical hacker?

The average salary for an Ethical Hacker in India is ₹51,251 approx.

Q4. What is the salary of an ethical hacker?

Get 90% Course fee refund on completing 90% course in 90 days! Take theThree 90 Challenge today.The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance toupskill and get 90% refund.What more motivation do you need?Start the challenge right away!

Q4. What is the salary of an ethical hacker?

R

Q4. What is the salary of an ethical hacker?

- Ethical Hacking

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon

- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial

- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Similar Reads

- System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management
- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL
- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn
- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]
- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025
- GATE CS & IT Course - 2025
- GATE DA Course 2025

Similar Reads

TypeScript Basics

- Introduction to TypeScript
- Difference between TypeScript and JavaScript
- How to install TypeScript ?
- Hello World in TypeScript
- How to execute TypeScript file using command line?
- Variables in TypeScript
- What are the different keywords to declare variables in TypeScript ?
- Identifiers and Keywords in TypeScript

TypeScript primitive types

- Data types in TypeScript
- TypeScript Numbers
- TypeScript String
- Explain the concept of null and its uses in TypeScript

TypeScript Object types

- What are TypeScript Interfaces?
- TypeScript class
- How enums works in TypeScript ?
- TypeScript Tuples

TypeScript other types

- What is any type, and when to use it in TypeScript ?
- How to Create an Object in TypeScript?
- What is an unknown type and when to use it in TypeScript ?
- Explain the purpose of never type in TypeScript

TypeScript combining types

- TypeScript Union
- What are type aliases and how to create it in Typescript ?

TypeScript Assertions

- Explain Type assertions in TypeScript

TypeScript Functions

- How to write a function in Typescript ?
- How to achieve function overloading in TypeScript ?
- Explain the arrow function syntax in TypeScript
- TypeScript toPrecision() Function
- TypeScript toFixed() Function
- TypeScript toLocaleString() Function
- TypeScript toString()

TypeScript interfaces and aliases

- What are TypeScript Interfaces?
- What are type aliases and how to create it in Typescript ?

TypeScript classes

- How to Extend an Interface from a class in TypeScript ?
- How to Create an Object in TypeScript?
- How to use getters/setters in TypeScript ?
- TypeScript Inheritance
- When to use interfaces and when to use classes in TypeScript ?
- Generics Interface in typescript
- How to use property decorators in TypeScript ?

TypeScript modules

- What are the Modules in Typescript ?
- How to import a module in Typescript ?

TypeScript Arrays

- TypeScript Arrays
- TypeScript Array push() Method
- TypeScript Array slice() Method
- TypeScript Array splice() Method
- TypeScript Array reverse() Method
- TypeScript Array reduce() Method
- TypeScript Array concat() Method
- TypeScript Array some() Method
- TypeScript Array shift() Method
- TypeScript Array map() Method
- TypeScript Array indexOf() Method
- Array filter() Method - TypeScript
- TypeScript Array reduceRight() Method
- TypeScript Array join() Method
- TypeScript Array unshift() Method
- TypeScript Array sort() Method
- TypeScript Array lastIndexOf() Method
- TypeScript Array pop() Method
- TypeScript Array unshift() Method
- TypeScript Array forEach() Method
- TypeScript Array toString() Method

Typescript String

- TypeScript String

- TypeScript | String Constructor Property
- TypeScript String toLowerCase() Method
- TypeScript String toString() Method
- TypeScript String toLocaleUpperCase() Method
- TypeScript String substring() Method
- TypeScript String substr() Method
- TypeScript String split() Method
- TypeScript | String valueOf() Method
- TypeScript String replace() Method
- TypeScript String toLocaleLowerCase() Method
- TypeScript | String lastIndexOf() Method
- TypeScript String slice()
- TypeScript String concat() Method
- TypeScript String indexOf() Method
- TypeScript String Length Property
- TypeScript String search() Method
- TypeScript String Prototype Property
- TypeScript String charCodeAt() Method

TypeScript Exercises

- TypeScript Exercises, Practice Questions and Solutions

TypeScript Tutorial

-
-
-

TypeScript Tutorial

TypeScript is a superset of JavaScript that adds static typing, interfaces, and enhanced features to JavaScript. In other words, we can say that TypeScript is JavaScript with added syntax for types.

TypeScript Tutorial

- Due to Static Typing, you can define variable types, helping to catch errors before running the code.
- It is compiled into JavaScript, ensuring compatibility across all JavaScript environments.
- It is widely used for web development on both the client and server sides.
- TypeScript reduces runtime errors with compile-time type checking.
- Microsoft developed it and it is trusted by companies like Google, Slack, Airbnb, and Asana.

Hello World Program

To start with TypeScript you need to install it in your project. Follow this article to install it on your system

Hello World Program

- Install TypeScript in your System
- Setup TypeScript Project

Hello World Program

Let us now take a look at our first code example

Hello World Program

It will print 'Hello, TypeScript' in the console.

Why to learn TypeScript?

TypeScript is increasingly becoming the go-to language for modern web development due to its advantages over plain JavaScript.

Why to learn TypeScript?

- Stronger Code:TypeScript adds types to JavaScript, reducing errors and improving code quality.
- Scalability:It is easier to maintain and scale larger applications, especially with complex codebases.
- Compatibility with JavaScript:TypeScript is a superset of JavaScript, meaning you can gradually adopt it in existing projects.
- Tooling Support:It is integrated with IDEs and build tools, offering debugging, code linting, and other features.
- Popular Frameworks Support:It is widely used with modern frameworks likeReact,Angular, andVuefor better type safety.
- Early Bug Detection:TypeScript helps catch potential bugs at compile-time, saving time during runtime testing.
- Growing Job Demand:TypeScript is in high demand among employers, especially for large-scale web and enterprise applications.

Fundamentals

- Introduction
- Variables
- DataTypes
- Type Annotations
- Type Inference
- Enums
- Literal Types
- Union and Intersection Types
- Type Aliases

Functions in TypeScript

- Function Types
- Call Signatures

- Optional and Default Parameters
- Rest Parameters
- Arrow Functions with TypeScript
- Function Overloading

Object Types and Interfaces

- Object Types Overview
- Interfaces
- Type Aliases vs Interfaces

Classes and Object-Oriented Programming

- Classes and Properties
- Access Modifiers
- Readonly Properties
- Getters and Setters
- Abstract Classes and Methods
- Implementing Interfaces in Classes

Generics

- Introduction to Generics
- Generic Functions
- Generic Classes
- Constraints in Generics
- Built-in Generic Types

Advanced Types

- Type Assertions
- Type Guards
- Conditional Types

- Mapped Types
- Template Literal Types
- Recursive Types
- Utility Types

Modules and Namespaces

- TypeScript Modules
- Default and Named Exports
- Legacy Namespaces

Testing with TypeScript

- Writing Unit Tests with TypeScript
- Testing Frameworks
- Type-safe Mocking

TypeScript Configuration and Performance

- Optimizing tsconfig.json for large projects
- Performance tuning for TypeScript compilation

TypeScript Libraries and Frameworks

TypeScript is widely used with popular frameworks and libraries:

TypeScript Libraries and Frameworks

- Frontend Frameworks: React, Angular, Vue
- Backend Frameworks: NestJS, Express.js, Koa
- State Management Libraries: Redux Toolkit, Zustand

TypeScript vs JavaScript

Feature | JavaScript | TypeScript

Typing | Dynamically typed | Statically typed

Error Detection | Errors caught at runtime | Errors caught at compile time

Tooling Support | Limited tooling support for type checking | Advanced tooling with autocomplete

Scalability | Suitable for small to medium projects | Ideal for large, complex applications

Compatibility | Runs natively in browsers and servers | Compiles to JavaScript for compatibility

TypeScript vs JavaScript

For more detailed comparison read the article-Difference between TypeScript and JavaScript

What is TypeScript?

TypeScript is a superset of JavaScript that adds static typing and modern features for building scalable applications.

How is TypeScript different from JavaScript?

TypeScript introduces type annotations and advanced features like interfaces, generics, and decorators, which JavaScript lacks.

How does TypeScript compile?

TypeScript compiles to plain JavaScript, ensuring compatibility with any browser, runtime, or JavaScript library.

Is TypeScript difficult to learn?

If you already know JavaScript, learning TypeScript is simple, as it builds on JavaScript's foundation and introduces types incrementally.

Can I use TypeScript with existing JavaScript projects?

Yes, TypeScript can be adopted incrementally. You can start by renaming .js files to .ts and adding type annotations.

What are some popular frameworks built with TypeScript?

Angular, NestJS, and Next.js are prominent frameworks built with TypeScript.

What are some popular frameworks built with TypeScript?

MasterDSA with JavaScript in just 90 days. Explore core DSA concepts, refine your coding skills, and tackle real-world challenges. Take on the Three 90 Challenge?complete 90% of the course in 90 days and earn a 90% refund as a reward for your commitment!

What are some popular frameworks built with TypeScript?

M

What are some popular frameworks built with TypeScript?

- JavaScript
- TypeScript
- Web Technologies
- Web-Tech Tutorials
- Tutorials

Similar Reads

Similar Reads

- Company
- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GFG Corporate Solution
- Placement Training Program

Similar Reads

- Explore
- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos
- Geeks Community

Similar Reads

- Languages
- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

Similar Reads

- DSA
- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems

- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

Similar Reads

- Data Science & ML
- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

Similar Reads

- Web Technologies
- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Similar Reads

- Python Tutorial
- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Similar Reads

- Computer Science
- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Similar Reads

- DevOps
- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP

- DevOps Roadmap

Similar Reads

- System Design
- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Similar Reads

- School Subjects
- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Similar Reads

- Commerce
- Accountancy
- Business Studies
- Economics
- Management

- HR Management
- Finance
- Income Tax

Similar Reads

- Databases
- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

Similar Reads

- Preparation Corner
- Company-Wise Recruitment Process
- Resume Templates
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation
- Companies
- Colleges

Similar Reads

- Competitive Exams
- JEE Advanced
- UGC NET
- UPSC
- SSC CGL

- SBI PO
- SBI Clerk
- IBPS PO
- IBPS Clerk

Similar Reads

- More Tutorials
- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets
- Recent Articles

Similar Reads

- Free Online Tools
- Typing Test
- Image Editor
- Code Formatters
- Code Converters
- Currency Converter
- Random Number Generator
- Random Password Generator

Similar Reads

- Write & Earn

- Write an Article
- Improve an Article
- Pick Topics to Write
- Share your Experiences
- Internships

Similar Reads

- DSA/Placements
- DSA - Self Paced Course
- DSA in JavaScript - Self Paced Course
- DSA in Python - Self Paced
- C Programming Course Online - Learn C with Data Structures
- Complete Interview Preparation
- Master Competitive Programming
- Core CS Subject for Interview Preparation
- Mastering System Design: LLD to HLD
- Tech Interview 101 - From DSA to System Design [LIVE]
- DSA to Development [HYBRID]
- Placement Preparation Crash Course [LIVE]

Similar Reads

- Development/Testing
- JavaScript Full Course
- React JS Course
- React Native Course
- Django Web Development Course
- Complete Bootstrap Course
- Full Stack Development - [LIVE]

- JAVA Backend Development - [LIVE]
- Complete Software Testing Course [LIVE]
- Android Mastery with Kotlin [LIVE]

Similar Reads

- Machine Learning/Data Science
- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Mastering Generative AI and ChatGPT
- Data Science Course with IBM Certification

Similar Reads

- Programming Languages
- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Similar Reads

- Clouds/Devops
- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Similar Reads

- GATE
- GATE CS & IT Test Series - 2025
- GATE DA Test Series 2025

- GATE CS & IT Course - 2025

- GATE DA Course 2025

Similar Reads