Node.js

TechGuide - Alura, FIAP e PM3

Node.js

Level 1

JavaScript - Fundamentals:

JavaScript is the world's most popular programming language and is one of the core technologies of the World Wide Web, alongside HTML and CSS. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.

Knowing the primitive types

Declaring variables, considering the difference between 'var', 'let' and 'const

Using conditional structures ('if', 'else')

Know the assignment and comparison operators ('=', '==', '===')

Using repetition structures and loops ('while', 'for')

Using functions, passing parameters and arguments

Manipulating arrays and lists

Getting data from an API

Making asynchronous calls using 'Async/Await', 'Promise', etc

Contents

WebSite W3Schools: JavaScript Syntax (https://www.w3schools.com/js/js_syntax.asp)

WebSite W3Schools: JavaScript Variables (https://www.w3schools.com/js/js_variables.asp)

WebSite W3Schools: JavaScript Operators (https://www.w3schools.com/js/js_operators.asp)

WebSite W3Schools: JavaScript Comparison and Logical Operators (https://www.w3schools.com/js/js_comparisons.asp)

WebSite W3Schools: JavaScript Data Types (https://www.w3schools.com/js/js_datatypes.asp)

WebSite W3Schools: JavaScript Arrays (https://www.w3schools.com/js/js_arrays.asp)

WebSite W3Schools: JavaScript if, else, and else if (https://www.w3schools.com/js/js_if_else.asp)

WebSite W3Schools: JavaScript Iterables (while, for) (https://www.w3schools.com/js/js_iterables.asp)

WebSite W3Schools: JavaScript Functions (https://www.w3schools.com/js/js_functions.asp)

WebSite W3Schools: JavaScript Objects (https://www.w3schools.com/js/js_objects.asp)

WebSite W3Schools: JavaScript Classes (https://www.w3schools.com/js/js_classes.asp)

WebSite MDN Web Docs: A first splash into JavaScript (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash)

WebSite MDN Web Docs: Handling text — strings in JavaScript (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings)

WebSite MDN Web Docs: if...else (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else)

WebSite MDN Web Docs: while (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while)

WebSite MDN Web Docs: for (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for)

WebSite MDN Web Docs: Arrays (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays)

WebSite MDN Web Docs: Template strings (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals)

Article An introduction to JavaScript's async and await (https://medium.com/@zellwk/an-introduction-to-javascripts-async-and-await-edb313356677)

Article Javascript: Map, Filter, and Reduce (https://harsh-patel.medium.com/javascript-map-filter-and-reduce-51da071d6dc4)

YouTube Aaron Jack: Learn JavaScript in just 5 minutes (https://www.youtube.com/watch?v=c-I5S_zTwAc)

YouTube Topknot Clare: JavaScript Basics in 10 Minutes (https://www.youtube.com/watch?v=c-I5S_zTwAc)

YouTube Programming with Mosh: JavaScript Tutorial for Beginners - Learn JavaScript in 1 Hour (https://www.youtube.com/watch?v=W6NZfCO5SIk)

YouTube freeCodeCamp.org: Introduction to JavaScript (https://www.youtube.com/watch?v=_y9oxzTGERs)

YouTube freeCodeCamp.org: Learn JavaScript - Full Course for Beginners (https://www.youtube.com/watch?v=PkZNo7MFNFg)

Node.js - Fundamentals:

Node.js is a JavaScript execution environment that allows you to run applications developed with the language autonomously, without depending on a browser.

Getting to know blocking and non-blocking operations

Learning the concept of event loops

Learning how to use Node.js libraries, such as 'net', 'fs', 'http', 'path', among others

Understanding how Timers work

Contents

WebSite Node.js - Documentation (https://nodejs.dev/en/learn/)

WebSite Node.js - Documentation - Overview of Blocking vs Non-Blocking (https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/)

WebSite Node.js - Documentation - Event Loop (https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/)

WebSite Node.js - Documentation - Timers (https://nodejs.org/en/docs/guides/timers-in-node/)

Article Introduction to Node.js — Create Your First Web App (https://betterprogramming.pub/introduction-to-node-js-create-your-first-web-app-96b705c28023)

Article Introduction to Node.js / A beginners guide to Node.js and NPM (https://medium.com/jspoint/introduction-to-node-js-a-beginners-guide-to-node-js-and-npm-eca9c408f9fe)

YouTube Fireship: Node.js Ultimate Beginner's Guide in 7 Easy Steps (https://www.youtube.com/watch?v=ENrzD9HAZK4)

YouTube freeCodeCamp.org: Node.js and Express.js - Full Course (https://www.youtube.com/watch?v=Oe421EPjeBE)

JavaScript - Callbacks and Promises:

A Promise is a proxy for a value not necessarily known when the promise is created. This lets asynchronous methods return values like synchronous methods - instead of immediately returning the final value, the asynchronous method returns a promise to supply the value at some point in the future.

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.

An async function is a function declared with the async keyword, and the await keyword is permitted within it. The async and await keywords enable asynchronous, promise-based behavior to be written in a cleaner style, avoiding the need to explicitly configure promise chains.

Understanding the concept of asynchronous programming

Writing asynchronous code understanding the concept of promises in JavaScript

Using JavaScript methods, keywords and objects for handling promises like 'Async/Await', '.then()', 'Promise', etc

Learning in which situations you need to use asynchronous programming

Calling APIs with `fetch()`

Contents

WebSite W3Schools: JavaScript Callbacks (https://www.w3schools.com/js/js_callback.asp)

WebSite W3Schools: JavaScript Promises (https://www.w3schools.com/js/js_promise.asp)

WebSite W3Schools: JavaScript Async (https://www.w3schools.com/js/js_async.asp)

WebSite W3Schools: Asynchronous JavaScript (https://www.w3schools.com/js/js_asynchronous.asp)

WebSite W3Schools: JavaScript Fetch API (https://www.w3schools.com/js/js_api_fetch.asp)

WebSite MDN Web Docs: Promise (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)

WebSite MDN Web Docs: Callback function (https://developer.mozilla.org/en-US/docs/Glossary/Callback_function)

WebSite MDN Web Docs: async function (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function)

WebSite MDN Web Docs: Introducing asynchronous JavaScript (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Introducing)

Article A practical ES6 guide on how to perform HTTP requests using the Fetch API (https://medium.com/free-code-camp/a-practical-es6-guide-on-how-to-perform-http-requests-using-the-fetch-api-594c3d91a547)

YouTube James Q Quick: JavaScript Callbacks Explained in 5 Minutes (https://www.youtube.com/watch?v=kz_vwAF4NHI)

YouTube Code with Ania Kubów: Callbacks in JavaScript Explained (https://www.youtube.com/watch?v=cNjIUSDnb9k)

YouTube Web Dev Simplified: JavaScript Promises In 10 Minutes (https://www.youtube.com/watch?v=DHvZLI7Db8E)

YouTube Traversy Media: Async JS Crash Course - Callbacks, Promises, Async Await (https://www.youtube.com/watch?v=PoRJizFvM7s)

JavaScript - Error handling:

Error handling refers to the response and recovery procedures from error conditions present in a software application. In other words, it is the process comprised of anticipation, detection and resolution of application, programming or communication errors.

Knowing and handling the most common exceptions

Knowing the types of errors and in which situations they can occur

Using 'try' and 'catch' for error handling

Learning on what occasions and how to use`throw`

Creating specific exceptions according to your application's needs

Contents

`WebSite`	MDN Web Docs: Control flow and error handling (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling#declara%C3%A7%C3%B5es_de
`Article`	Error Handling in JavaScript: a Quick Guide (https://medium.com/swlh/error-handling-in-javascript-a-quick-guide-54b954427e47)
`Article`	Better error handling in JavaScript (https://iaincollins.medium.com/error-handling-in-javascript-a6172ccdf9af)
`Article`	JavaScript Best Practices — Error Handling Techniques (https://medium.com/swlh/javascript-best-practices-error-handling-techniques-20a8752d92f6)
`Article`	JavaScript Clean Code: Error Handling (https://betterprogramming.pub/javascript-clean-code-error-handling-9cd7e87fdbe3)
`Article`	JavaScript error handling best practices (https://medium.com/@raygunio/javascript-error-handling-best-practices-ac5d4e667e09)
`Article`	JavaScript Error Handling from Express.js to React (https://itnext.io/javascript-error-handling-from-express-js-to-react-810deb5e5e28)
`YouTube`	freeCodeCamp.org: try, catch, finally, throw - error handling in JavaScript (https://www.youtube.com/watch?v=cFTFtuEQ-10)
`YouTube`	dcode: Basics of Error Handling in JavaScript (https://www.youtube.com/watch?v=G8Jux_bsIXU)
`YouTube`	Dave Gray: Javascript Error Handling (https://www.youtube.com/watch?v=blBoIyNhGvY)

Object-oriented Programming Concepts:

Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.

How objects work

Creating and using constructors

What classes are

Creating and using Methods

How encapsulation works

What inheritance is

What polymorphism is

How interfaces work

What abstractions are

Contents

WebSite MDN Web Docs: Object-oriented programming (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming)

WebSite Geeks for Geeks: Object Oriented Programming (OOPs) Concept in Java (https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/)

WebSite W3Schools: Object-Oriented Programming Style (https://www.w3schools.blog/object-oriented-programming-style)

Article What is object-oriented programming? OOP explained in depth (https://www.educative.io/blog/object-oriented-programming)

YouTube Programming with Mosh: Object-oriented Programming in 7 minutes (https://www.youtube.com/watch?v=QJjY2TNyl-8)

YouTube freeCodeCamp.org: Intro to Object Oriented Programming - Crash Course (https://www.youtube.com/watch?v=SiBw7os-_zI)

YouTube Traversy Media: JavaScript OOP Crash Course (ES5 & ES6) (https://www.youtube.com/watch?v=vDJpGenyHaA)

Node.js - Testing:

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Using unit tests

Using integration testing

Using behavioral testing

Using mocks

Learn the concepts of BDD (Behavior-Driven Development) and TDD (Test-Driven Development)

Contents

Article Unit Testing in Node.js: The Basics and a Quick Tutorial (https://brightsec.com/blog/unit-testing-in-nodejs/)

Article A Node.js Guide to Actually Doing Integration Tests (https://www.toptal.com/nodejs/nodejs-guide-integration-tests)

Article Revisiting Node.js Testing: Part 1 (https://codeburst.io/revisiting-node-js-testing-part-1-84c33bb4d711)

YouTube Sam Meech-Ward: Testing Node Server with Jest and Supertest (https://www.youtube.com/watch?v=FKnzS_icp20)

Article Marius Espejo: NodeJS Express Test-Driven API Development (TDD) (https://www.youtube.com/watch?v=M44umyYPiuo)

Article MammothInteractive: Simple Node.js Unit Testing with Mocha (https://www.youtube.com/watch?v=xDI2KK2gSgQ)

Level 2

SOLID:

SOLID has five principles that are considered best practices in software development that help programmers write cleaner code by separating responsibilities, reducing docking, easing refactoring, and encouraging code reuse.

Contents

Article The S.O.L.I.D Principles in Pictures (https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898)

Article SOLID Principles every Developer Should Know (https://blog.bitsrc.io/solid-principles-every-developer-should-know-b3bfa96bb688)

YouTube Fireship: Solid in 100 Seconds (https://www.youtube.com/watch?v=q0BGgQJcp7w)

YouTube A Dev' Story: SOLID Design Principles Explained in a Nutshell (https://www.youtube.com/watch?v=69sfWNzxTMc)

YouTube Web Dev Simplified: Single Responsibility Principle Explained - SOLID Design Principles (https://www.youtube.com/watch?v=UQqY3_6Epbg)

Node.js - Express:

Express is a flexible Node.js web application framework that provides a robust feature set for web and mobile applications.

Using the Express framework to build REST APIs with Node.js

Managing requests for different HTTP verbs on different URLs

Defining ports that will be used for connection and the location of the templates that are used to render the response

Creating route handlers using the 'router' method

Getting to know the 'Router' libraries and their HTTP verbs, such as 'get', 'post', 'put', etc

Defining endpoints with route paths

Contents

WebSite Express.js - Documentation (https://expressjs.com/)

WebSite Express.js - Documentation - Routing (https://expressjs.com/en/guide/routing.html)

Article MDN Web Docs: Introduction to Express/Node.js (https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction#introducing_express)

Article Going out to eat and understanding the basics of Express.js (https://medium.com/free-code-camp/going-out-to-eat-and-understanding-the-basics-of-express-js-f034a029fb66)

Article Build A RESTful Api With Node.js And Express.Js (https://medium.com/@haybams/build-a-restful-api-with-node-js-and-express-js-d7e59c7a3dfb)

Article Understanding Express.js: Creating Your Own Node HTTP Request Router (https://medium.com/hackernoon/understanding-express-js-creating-your-own-node-http-request-router-4190a9b6aad6)

YouTube Web Dev Simplified: Learn Express JS In 35 Minutes (https://www.youtube.com/watch?v=SccSCuHhOw0)

YouTube Traversy Media: Express JS Crash Course (https://www.youtube.com/watch?v=L72fhGm1tfE)

Node.js - ORM:

Object-Relational Mapping (ORM) is a technique used to map object-oriented systems to relational databases, where the database tables are represented in classes and the table records would be instances of these classes.

Understanding what ORMs are and what they are used for

Knowing SQL and its database managers

Working with Sequelize, an ORM used with Node.js

Knowing other Node.js ORMs, like Prisma

Contents

`WebSite` Sequelize - Documentation (https://sequelize.org/)

`Article` What is an ORM and Why You Should Use it (https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a)

`Article` The Ultimate Guide To Get Started With Sequelize ORM (https://levelup.gitconnected.com/the-ultimate-guide-to-get-started-with-sequelize-orm-238588d3516e)

`Article` Adding Sequelize To Your Node.js App (https://medium.com/mtholla/integrating-sequelize-into-your-node-app-a446353fb5ee)

`YouTube` PedroTech: How to Use Sequelize ORM in NodeJS - Tutorial (https://www.youtube.com/watch?v=Crk_5Xy8GMA)

`YouTube` Traversy Media: Build a Node.js App With Sequelize (https://www.youtube.com/watch?v=tF-ly2sB3hU)

`YouTube` Adnan Halilovic: Learn Prisma MySQL CRUD with NodeJS in simple and easy steps! (https://www.youtube.com/watch?v=mmjQ280wnlw)

Node.js - Authentication and Tokens:

JSON Web Tokens (JWT) are an open, industry standard RFC 7519 method for representing claims securely between two parties.

Building an authentication system using tokens

Understanding how JSON Web Tokens (JWT) work

Building an allowlist to store opaque tokens

Implementing methods for updating tokens

Contents

`WebSite` JWT - Documentation (https://jwt.io/)

`Article` Authentication and Authorization using JWT with Node.js (https://betterprogramming.pub/authentication-and-authorization-using-jwt-with-node-js-4099b2e6ca1f)

`Article` Building a RESTful CRUD API with Node.js, JWT, Bcrypt, Express and MongoDB (https://medium.com/zero-equals-false/building-a-restful-crud-api-with-node-js-jwt-bcrypt-express-and-mongodb-4e1fb20b7f3d)

`YouTube` Web Dev Simplified: JWT Authentication Tutorial - Node.js (https://www.youtube.com/watch?v=mbsmsi7l3r4)

`YouTube` Ben Tedder: Create a Node API with JWT's (https://www.youtube.com/watch?v=xBYr9DxDqyU)

TypeScript - Fundamentals:

TypeScript is a strongly typed programming language that builds on JavaScript.

Understanding in depth what types are and the importance of typed programming

Learning what TypeScript is, why it was created, how it works and its relationship with JavaScript

Knowing the TypeScript tools (integration with the code editor, static checker and compiler)

Writing code in TypeScript using its tools (interfaces, enum, decorators, etc.)

Contents

WebSite W3Schools: TypeScript Introduction (https://www.w3schools.com/typescript/typescript_intro.php)

WebSite TypeScript Documentation (https://www.typescriptlang.org/docs/)

WebSite Documentação: TypeScript for the New Programmer (https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html)

WebSite TypeScript - Understanding TypeScript (https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/january/typescript-understanding-typescript)

Article W3Schools: Differences Between JavaScript And TypeScript (https://www.w3schools.blog/differences-between-javascript-and-typescript)

Article W3Schools: TypeScript Tutorial: A step-by-step guide to learn TypeScript (https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8)

YouTube Programming with Mosh: TypeScript Tutorial for Beginners (https://www.youtube.com/watch?v=d56mG7DezGs)

YouTube Academind: TypeScript Course for Beginners - Learn TypeScript from Scratch (https://www.youtube.com/watch?v=BwuLxPH8IDs)

Level 3

Nest.js - Fundamentals:

NestJS is a Node framework with full TypeScript support that runs on top of HTTP frameworks like ExpressJS or Fastify. It uses many elements of object-oriented programming and a lot of TypeScript functionality.

Learning what NestJS is and why it is used

Using specific NestJS features, such as providers, modules and controllers

Developing APIs using NestJS

Using the Nest.js CLI

Contents

WebSite NestJS - Documentation (https://docs.nestjs.com/)

Article Nest.js - Introduction (https://dev.to/angelmtztrc/nest-js-introduction-part-1-1k21)

Article Controllers the NEST.js way (https://dev.to/xamhans/part-1-controllers-the-nest-js-way-3aab)

Article Dependency Injection in Nest.js (https://dev.to/krishnakurtakoti/dependency-injection-in-nest-js-2a7j)

YouTube Fireship: NestJS in 100 Seconds (https://www.youtube.com/watch?v=0M8AYU_hPas)

YouTube Academind: Learn Nest.js from Scratch by building an API (https://www.youtube.com/watch?v=F_oOtaxb0L8)

Microservices architecture:

Microservices are an architectural approach in which software consists of small independent services that communicate with each other and are organized according to their business domains.

Learning the concept of planned architecture for microservices

Performing communication using APIs

Improving the scalability of a system

Contents

WebSite Google Cloud: Microservice architecture style (https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices)

WebSite Amazon: What are microservices? (https://aws.amazon.com/en/microservices/)

Article The What, Why, and How of a Microservices Architecture (https://medium.com/hashmapinc/the-what-why-and-how-of-a-microservices-architecture-4179579423a9)

Article Understanding Microservices: From Idea To Starting Line (https://medium.com/free-code-camp/microservices-from-idea-to-starting-line-ae5317a6ff02)

YouTube TechWorld with Nana: Microservices explained - the What, Why and How? (https://www.youtube.com/watch?v=rv4LlmLmVWk)

YouTube ByteByteGo: What Are Microservices Really All About? (And When Not To Use It) (https://www.youtube.com/watch?v=_Oyy5PFOIcU)

YouTube InfoQ: Mastering Chaos - A Netflix Guide to Microservices (https://www.youtube.com/watch?v=CZ3wIuvmHeM)

YouTube freeCodeCamp.org: Microservice Architecture and System Design with Python & Kubernetes – Full Course (https://www.youtube.com/watch?v=hmkF77F9TLw)

Containers:

Containers are software packages that contain all the elements needed to run in any environment.

Kubernetes (also known as k8s or “kube”) is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.

Isolating your software to run independently

Deploying software in clusters

Modularizing your system into smaller packages

Getting to know the Docker platform

Getting to know Kubernetes

Contents

WebSite IBM: Containers (https://www.ibm.com/en-us/cloud/learn/containers)

WebSite HPE: What are containers? (https://www.hpe.com/us/en/what-is/containers.html)

WebSite Google Cloud: What are Containers? (https://cloud.google.com/learn/what-are-containers)

WebSite Azure: What is a container? (https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-container/)

WebSite Kubernetes - Documentation (https://kubernetes.io/docs/tutorials/kubernetes-basics/)

Article Red Hat: What is Kubernetes? (https://www.redhat.com/en/topics/containers/what-is-kubernetes)

Article Azure: What is Kubernetes? (https://azure.microsoft.com/en-us/topic/what-is-kubernetes/)

YouTube freeCodeCamp.org: Docker Tutorial for Beginners (https://www.youtube.com/watch?v=eLiHH9Bu04E)

YouTube freeCodeCamp.org: Docker Containers and Kubernetes Fundamentals – Full Hands-On Course (https://www.youtube.com/watch?v=kTp5xUtcalw)

YouTube TechWorld with Nana: Kubernetes Tutorial for Beginners [FULL COURSE in 4 Hours] (https://www.youtube.com/watch?v=X48VuDVv0do)

WebSockets:

The WebSocket API is an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

Getting to know the WebSocket protocol and its use in client-server communication

Learning about the various uses of WebSockets on the web

Creating applications that use WebSockets with Node.js APIs and libraries

Contents

WebSite MDN Web Docs: The WebSocket API (WebSockets) (https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)

Article An Introduction To WebSocket (https://gaierken.medium.com/an-introduction-to-websocket-f4a30024b7f2)

Article A Brief Introduction to WebSocket (https://javascript.plainenglish.io/a-brief-introduction-to-websocket-eada28567057)

Article Introduction to WebSockets with Socket.io in Node.js (https://towardsdev.com/introduction-to-websockets-with-socket-io-in-node-js-f5bc4a693b24)

Article How To Create a WebSocket in Python (https://betterprogramming.pub/how-to-create-a-websocket-in-python-b68d65dbd549)

Article Node.js & WebSocket — Simple chat tutorial (https://medium.com/@martin.sikora/node-js-websocket-simple-chat-tutorial-2def3a841b61)

YouTube Fireship: WebSockets in 100 Seconds & Beyond with Socket.io (https://www.youtube.com/watch?v=1BfCnjr_Vjg)

YouTube freeCodeCamp.org: A Beginner's Guide to WebSockets (https://www.youtube.com/watch?v=8ARodQ4Wlf4)

YouTube Be A Better Dev: REST API (HTTP) vs Websockets - Concept Overview With Example (https://www.youtube.com/watch?v=fG4dkrlaZAA)

GraphQL:

GraphQL is a new API standard that provides a more efficient, powerful and flexible alternative to REST. It was developed and open-sourced by Facebook and is now maintained by a large community of companies and individuals from all over the world.

Understanding how GraphQL is used in API development

Creating APIs using GraphQL libraries and frameworks

Contents

WebSiteGraphQL: Documentation (https://graphql.org/)

WebSiteHow to GraphQL: The Fullstack tutorial for GraphQL (https://www.howtographql.com/)

WebSiteRedHat: What is GraphQL? (https://www.redhat.com/en/topics/api/what-is-graphql)

Article So what's this GraphQL thing I keep hearing about? (https://medium.com/free-code-camp/so-whats-this-graphql-thing-i-keep-hearing-about-baf4d36c20cf)

Article Why GraphQL is the future of APIs (https://medium.com/free-code-camp/why-graphql-is-the-future-of-apis-6a900fb0bc81)

Article GraphQL: A Success Story for PayPal Checkout (https://medium.com/paypal-tech/graphql-a-success-story-for-paypal-checkout-3482f724fb53)

Article GraphQL: Everything You Need to Know (https://blog.weblab.technology/graphql-everything-you-need-to-know-58756ff253d8)

Article REST vs. GraphQL: A Critical Review (https://medium.com/good-api/rest-vs-graphql-a-critical-review-5f77392658e7)

YouTube Web Dev Simplified: Learn GraphQL In 40 Minutes (https://www.youtube.com/watch?v=ZQL7tL2S0oQ)

YouTube The Net Ninja: GraphQL Tutorial (https://www.youtube.com/watch?v=Y0lDGjwRYKw&list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f)

YouTube Fireship: GraphQL Explained in 100 Seconds (https://www.youtube.com/watch?v=eIQh02xuVw4)

Apollo Client:

Apollo Client is a comprehensive state management library for JavaScript that enables you to manage both local and remote data with GraphQL.

Contents

WebSite Documentation: Apollo Client (https://www.apollographql.com/docs/react/)

Article Oh Hello Apollo Client, Goodbye Redux! (https://kulkarniankita9.medium.com/oh-hello-apollo-client-goodbye-redux-49dfdeda16d1)

Article Apollo Client, now with React Hooks (https://www.apollographql.com/blog/announcement/frontend/apollo-client-now-with-react-hooks/)

Article Angular & Apollo Client: Getting Started with GraphQL in Angular (https://www.bitovi.com/blog/angular-and-apollo-client-get-started-with-graphql-in-angluar)

YouTube PedroTech: GraphQL With React Tutorial - Apollo Client (https://www.youtube.com/watch?v=YyUWW04HwKY)

YouTube Fireship: GraphQL with Apollo Server 2.0 (https://www.youtube.com/watch?v=8D9XnnjFGMs)

YouTube Laith Academy: React With GraphQL (Apollo Client) Crash Course (https://www.youtube.com/watch?v=gAbIQx26wSI)

Auxiliary Skill: Infrastructure and good practices

Git & GitHub - Fundamentals:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub is a hosting service for software development and version control using Git.

Creating a repository

Cloning a repository

Committing, pushing and pulling to and from the repository

Reversing a commit

Creating branches and pull requests

Handling merge and conflicts

Contents

WebSite Git Reference Book (https://git-scm.com/book/en/v2)

WebSite GitHub Documentation (https://docs.github.com/en)

WebSite GitHub Pages Documentation (https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages)

WebSite W3Schools: Git Tutorial (https://www.w3schools.com/git/default.asp?remote=github)

WebSite Git School - Visualizing Git (https://git-school.github.io/visualizing-git/)

WebSite Dangit, Git!?! (https://dangitgit.com/)

Article Git Tutorial - Explore The Commands And Operations In Git (https://medium.com/edureka/git-tutorial-da652b566ece)

Article Git and Github Quickstart Tutorial (https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca)

Article Getting Started with Git and GitHub: A Complete Tutorial for Beginner (https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d)

YouTube Programming with Mosh: Git Tutorial for Beginners - Learn Git in 1 Hour (https://www.youtube.com/watch?v=8JJ101D3knE)

YouTube freeCodeCamp.org: Git and GitHub for Beginners - Crash Course (https://www.youtube.com/watch?v=RGOj5yH7evk)

YouTube Kevin Stratvert: Git and GitHub for Beginners Tutorial (https://www.youtube.com/watch?v=tRZGeaHPoaw)

YouTube Tech With Tim: Git Tutorial for Beginners - Git & GitHub Fundamentals In Depth (https://www.youtube.com/watch?v=DVRQoVRzMIY)

HTTP - Fundamentals:

HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.

Understanding the difference between HTTP verbs

Testing requests and checking the status codes in the browser

Learning how to make a HTTP request on the command line with WGET

Downloading an image with WGET

Performing a POST

Contents

WebSite W3Schools: What is HTTP? (https://www.w3schools.com/whatis/whatis_http.asp)

WebSite MDN Web Docs: An overview of HTTP (https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)

WebSite MDN Web Docs: HTTP request methods (https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods)

WebSite HTTP Cats (https://http.cat/)

WebSite HTTP Dogs (https://http.dog/)

Article HTTP codes as Valentine's Day comics (https://medium.com/@hanilim/http-codes-as-valentines-day-comics-8c03c805faa0)

Article	Here Are the most popular ways to make an HTTP request in JavaScript (	https://medium.com/free-code-camp/here-is-the-most-popular-ways-to-make-an-http-request-in-javascript-954ce8c95aaa	)
YouTube	Traversy Media: HTTP Crash Course & Exploration (	https://www.youtube.com/watch?v=iYM2zFP3Zn0	)
YouTube	Curious Code: HTTP Request Methods - GET, POST, PUT, DELETE (	https://www.youtube.com/watch?v=tkfVQK6UxDI	)
YouTube	freeCodeCamp.org: Postman Beginner's Course - API Testing (	https://www.youtube.com/watch?v=VywxIQ2ZXw4	)

Design Patterns:

In software engineering, a Design Pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is a description or template for how to solve a problem that can be used in many different situations. Design Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

Getting familiarized with and applying the main Design Patterns

Contents

WebSite	What's a design pattern? (	https://refactoring.guru/design-patterns/what-is-pattern	)
Article	Design Patterns — Introduction (	https://henriquesd.medium.com/design-patterns-introduction-220f811db857	)
Article	What Are Design Patterns and Do I Need Them? (	https://www.developer.com/design/what-are-design-patterns-and-do-i-need-them/	)
Article	The 3 Types of Design Patterns All Developers Should Know (with code examples of each) (	https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-need-to-know/	)
Article	10 Design Patterns every Software Architect and Software Engineer must know (	https://ravindraelicherla.medium.com/10-design-patterns-every-software-architect-must-know-b33237bc01c2	)
Article	Modern-Day Architecture Design Patterns for Software Professionals (	https://betterprogramming.pub/modern-day-architecture-design-patterns-for-software-professionals-9056ee1ed977	)
YouTube	Web Dev Simplified: Design Patterns (Playlist) (	https://www.youtube.com/playlist?list=PLZlA0Gpn_vH_CthENcPCM0Dww6a5XYC7f	)
YouTube	Programming with Mosh: Design Patterns in Plain English (	https://www.youtube.com/watch?v=NU_1StN5Tkk	)
YouTube	Fireship: 10 Design Patterns Explained in 10 Minutes (	https://www.youtube.com/watch?v=tv-_1er1mWI	)
YouTube	Traversy Media: 5 Design Patterns Every Engineer Should Know (	https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg	)

Command Line - Fundamentals:

CLI is a command line program that accepts text input to execute operating system functions.

Knowing the most important commands

Contents

WebSite	W3Schools: What is Command Line Interface (CLI)? (	https://www.w3schools.com/whatis/whatis_cli.asp	)

WebSite Microsoft Docs: Using command line arguments for Windows Terminal (https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows)

Article Advanced CLI: Commands You Should Know as a Developer (https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-7bc48c752a5e)

YouTube freeCodeCamp.org: Command Line Crash Course (https://www.youtube.com/watch?v=yz7nYlnXLfE)

YouTube Traversy Media: Command Line Crash Course For Beginners - Terminal Commands (https://www.youtube.com/watch?v=uwAqEzhyjtw)

Cloud - Fundamentals:

Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.

Knowing the difference between IaaS, PaaS and SaaS

Knowing the largest cloud providers

Specializing in a specific provider of your choice

Contents

WebSite Microsoft Azure: What is cloud computing? (https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/)

WebSite Amazon AWS: What is cloud computing? (https://aws.amazon.com/en/what-is-cloud-computing/)

Article A beginner's guide to the basics of what cloud computing is about (https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/)

Article Cloud Computing for Beginners (https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/)

Article What are Cloud Computing Services [IaaS, CaaS, PaaS, FaaS, SaaS] (https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e)

YouTube Simplilearn: Cloud Computing Tutorial for Beginners (https://www.youtube.com/watch?v=RWgW-CgdIk0)

YouTube Amazon Web Services: What is Cloud Computing? - Amazon Web Services (https://www.youtube.com/watch?v=mxT233EdY5c)

YouTube Ecourse Review: Cloud Computing Services Models - IaaS PaaS SaaS Explained (https://www.youtube.com/watch?v=36zducUX16w/)

Auxiliary Skill: Front-end

HTML - Fundamentals:

HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

Learning which tags are required for basic HTML

Creating a text paragraph

Displaying an image

Knowing the difference between 'h1', 'h2', 'h3', etc.

Creating a hyperlinked text

Creating a form with relevant fields

Creating an ordered or unordered list of items

Creating a list of items within a dropdown list

Linking to a CSS file

Creating a table

Adding IDs and classes

Contents

WebSite W3Schools: HTML Introduction (https://www.w3schools.com/html/html_intro.asp)

WebSite W3Schools: HTML Elements (https://www.w3schools.com/html/html_elements.asp)

WebSite W3Schools: HTML Formatting Elements (https://www.w3schools.com/html/html_formatting.asp)

WebSite W3Schools: HTML Block and Inline Elements (https://www.w3schools.com/html/html_blocks.asp)

WebSite W3Schools: HTML Layout Elements (https://www.w3schools.com/html/html_layout.asp)

WebSite W3Schools: HTML Lists (https://www.w3schools.com/html/html_lists.asp)

WebSite W3Schools: HTML Tables (https://www.w3schools.com/html/html_tables.asp)

WebSite W3Schools: HTML Forms (https://www.w3schools.com/html/html_forms.asp)

WebSite MDN Web Docs: HTML basics (https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)

Article HTML vs CSS: What’s the Difference? (https://learn.onemonth.com/html-vs-css/)

Article Getting Started with HTML (https://towardsdatascience.com/getting-started-with-html-98866c3e5c9e)

YouTube Programming with Mosh: HTML Tutorial for Beginners - HTML Crash Course (https://www.youtube.com/watch?v=qz0aGYrrlhU)

YouTube freeCodeCamp.org: Learn HTML – Full Tutorial for Beginners (https://www.youtube.com/watch?v=kUMe1FH4CHE)

YouTube SuperSimpleDev: HTML & CSS Full Course - Beginner to Pro (https://www.youtube.com/watch?v=G3e-cpL7ofc)

CSS - Fundamentals:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS can be used for very basic document text styling — for example, for changing the color and size of headings and links. It can be used to create a layout — for example, turning a single column of text into a layout with a main content area and a sidebar for related information. It can even be used for effects such as animation.

Learning the visual structure of a page, with 'margin' and 'padding'

Establishing the size with 'width' and 'height'

Learning about the position of an element ('static', 'relative' or 'absolute')

Learning about the display of an element ('block', 'inline', 'inline-block')

Learning how to position images in relation to text

Learning about alignment

Learning about font style

Learning the differences and advantages of using the different units of measurement in CSS (%, relative, etc)

Connecting to the elements (IDs, classes) of an HTML file

Changing the characteristics of an element when the mouse hovers over it

Learning box-sizing

Learning Flexbox

Learning Grid

Contents

`WebSite`W3Schools: CSS Tutorial (https://www.w3schools.com/css/)

`WebSite`W3Schools: CSS Introduction (https://www.w3schools.com/css/css_intro.asp)

`WebSite`W3Schools: CSS Syntax (https://www.w3schools.com/css/css_syntax.asp)

`WebSite`W3Schools: CSS Selectors (https://www.w3schools.com/css/css_selectors.asp)

`WebSite`W3Schools: CSS Box Model (https://www.w3schools.com/css/css_boxmodel.asp)

`WebSite`W3Schools: CSS Units (https://www.w3schools.com/css/css_units.asp)

`WebSite`MDN Web Docs: Getting started with CSS (https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started)

`WebSite`MDN Web Docs: CSS selectors (https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors)

`WebSite`MDN Web Docs: Cascade and inheritance (https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)

`WebSite`MDN Web Docs: display property (https://developer.mozilla.org/en-US/docs/Web/CSS/display)

`WebSite`MDN Web Docs: Basic concepts of Flexbox (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)

`WebSite`MDN Web Docs: Basic concepts of Grid Layout (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

`Article`CSS Tutorial: Learn the basics of CSS in 5 minutes (https://medium.com/free-code-camp/get-started-with-css-in-5-minutes-e0804813fc3e)

`Article`The Complete CSS Flex Tutorial (https://jst.hashnode.dev/complete-css-flex-tutorial)

`Article`The Complete CSS Grid Tutorial (https://jst.hashnode.dev/css-grid-tutorial)

`YouTube`Web Dev Simplified: Learn CSS in 20 Minutes (https://www.youtube.com/watch?v=1PnVor36_40)

`YouTube`freeCodeCamp.org: CSS Tutorial - Zero to Hero (Complete Course) (https://www.youtube.com/watch?v=1Rs2ND1ryYc)

`YouTube`SuperSimpleDev: HTML & CSS Full Course - Beginner to Pro (https://www.youtube.com/watch?v=G3e-cpL7ofc)

DOM - Fundamentals:

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

Understanding how the DOM tree works

Accessing and manipulating HTML and CSS elements

Accessing the parents and children of an element

Inserting a new element to the tree

Removing an element from the tree

Waiting for an event on a certain page element using addEventListener()

Contents

WebSite	W3Schools: JavaScript HTML DOM Methods (	https://www.w3schools.com/js/js_htmldom_methods.asp)
WebSite	W3Schools: JavaScript HTML DOM Elements (	https://www.w3schools.com/js/js_htmldom_elements.asp)
WebSite	W3Schools: JavaScript HTML DOM - Changing HTML (	https://www.w3schools.com/js/js_htmldom_html.asp)
WebSite	W3Schools: JavaScript HTML DOM - Changing CSS (	https://www.w3schools.com/js/js_htmldom_css.asp)
WebSite	W3Schools: JavaScript HTML DOM Events (	https://www.w3schools.com/js/js_htmldom_events.asp)
WebSite	W3Schools: JavaScript HTML DOM EventListener (	https://www.w3schools.com/js/js_htmldom_eventlistener.asp)
WebSite	W3Schools: JavaScript HTML DOM Elements (Nodes) (	https://www.w3schools.com/js/js_htmldom_nodes.asp)
WebSite	MDN Web Docs: Introdução ao DOM (	https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
WebSite	MDN Web Docs: querySelectorAll() (	https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll)
WebSite	MDN Web Docs: addEventListener() (	https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener)
Article	An introduction to the JavaScript DOM (	https://medium.com/free-code-camp/an-introduction-to-the-javascript-dom-512463dd62ec)
Article	The DOM of Javascript (	https://javascript.plainenglish.io/the-dom-of-javascript-848506ebf386)
YouTube	Zac Gordon: An Introduction to the DOM (Document Object Model) in JavaScript (	https://www.youtube.com/watch?v=l-0nPnSvbX8)
YouTube	Zac Gordon: An Introduction to DOM Events with JavaScript (	https://www.youtube.com/watch?v=QE1YQnhntgw)
YouTube	Web Dev Simplified: Learn DOM Manipulation In 18 Minutes (	https://www.youtube.com/watch?v=y17RuWkWdn8)

Javascript - Accessibility:

Web accessibility is the inclusive practice of ensuring there are no barriers that prevent interaction with, or access to, websites by people with physical disabilities, situational disabilities, and socio-economic restrictions on bandwidth and speed.

Writing code with accessibility in mind

Contents

WebSiteW3: Accessibility (https://www.w3.org/standards/webdesign/accessibility)

WebSite MDN Web Docs: CSS and JavaScript accessibility best practices (https://developer.mozilla.org/en-US/docs/Learn/Accessibility/CSS_and_JavaScript)

WebSite WebAIM: Accessible JavaScript (https://webaim.org/techniques/javascript/)

Article Writing JavaScript with accessibility in mind (https://medium.com/@matuzo/writing-javascript-with-accessibility-in-mind-a1f6a5f467b9)

YouTube Flux Academy: Accessible Web Design - What Is It & How To Do It (https://www.youtube.com/watch?v=-ao_Kc_8rpE)

YouTube SFHTML5: Accessibility Fundamentals with Rob Dodson (https://www.youtube.com/watch?v=z8xUCzToff8)

YouTube Fullstack Academy: Web Accessibility Guidelines - Accessibility tips for JavaScript Developers (https://www.youtube.com/watch?v=z8xUCzToff8)

YouTube freeCodeCamp.org: Web App Tutorial - JavaScript, Mobile First, Accessibility, Persistent Data, Sass (https://www.youtube.com/watch?v=z8xUCzToff8)

Course Web.dev: Learn Accessibility! (https://web.dev/learn/accessibility/welcome)

SEO Strategies:

Search engine optimization (SEO) is the process of improving the quality and quantity of website traffic to a website or a web page from search engines.

Choosing keywords

Understanding how Google ranks pages

Knowing the ranking factors

Performing Link Building

Contents

WebSite Google Search Central: Search Engine Optimization (SEO) Starter Guide (https://developers.google.com/search/docs/fundamentals/seo-starter-guide)

Article Moz: The Beginner's Guide to SEO (https://moz.com/beginners-guide-to-seo)

Article Neil Patel: What is SEO? Your Complete Step-By-Step Guide (https://neilpatel.com/what-is-seo/)

YouTube Simplilearn: SEO In 5 Minutes | What Is SEO And How Does It Work (https://www.youtube.com/watch?v=MYE6T_gd7H0)

YouTube Fireship: SEO for Developers in 100 Seconds (https://www.youtube.com/watch?v=-B58GgsehKQ)

YouTube Ahrefs: Complete SEO Course for Beginners - Learn to Rank #1 in Google (https://www.youtube.com/watch?v=xsVTqzratPs)

TechGuide - Alura
Alura, PM3 e FIAP
O Techguide.sh é um projeto open source