React

TechGuide - Alura, FIAP e PM3

Level 1

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)

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-l5S_zTwAc)

- `YouTube` Topknot Clare: JavaScript Basics in 10 Minutes (https://www.youtube.com/watch?v=c-l5S_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)

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=I-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)

SPA Concepts:

- A single-page application (SPA) is a web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server, instead of the default method of a web browser loading entire new pages. The goal is faster transitions that make the website feel more like a native app.
- Understanding what an SPA is
- Establishing routes to other pages
- Knowing SPA frameworks
- Communicating with APIs

Contents

- `WebSite` Wikipédia: Single-page application (SPA) (https://en.wikipedia.org/wiki/Single-page_application)
- `WebSite` MDN Web Docs: SPA (Single-page application) (https://developer.mozilla.org/en-US/docs/Glossary/SPA)
- `Article` What I wish I had known about SPAs (https://stackoverflow.blog/2021/12/28/what-i-wish-i-had-known-about-single-page-applications/)
- `Article` Rise of the SPA (https://writing.pupius.co.uk/rise-of-the-spa-fb44da86dc1f)
- `Article` JavaScript: Vanilla Single Page Applications (SPA) (https://betterprogramming.pub/js-vanilla-script-spa-1b29b43ea475)
- `Article` Learn React JS — Build a Portfolio Single Page Application (SPA) (https://codeburst.io/learn-react-js-build-a-portfolio-single-page-application-spa-

ba001082a711)

- **YouTube** Academind: Dynamic Websites vs Static Pages vs Single Page Apps (SPAs) (https://www.youtube.com/watch?v=Kg0Q_YaQ3Gk)

- **YouTube** dcode: Build a Single Page Application with JavaScript (No Frameworks) (https://www.youtube.com/watch?v=6BozpmSjk-Y)

React - Components:

- React lets you define components as classes or functions. Components defined as classes provide more features. They accept arbitrary inputs (called "props") and return React elements describing what should appear on the screen.

- Understanding how components work

- Knowing the Styled Components library

- Understanding the difference between class and functional components

Contents

- **WebSite** W3Schools: React Components (https://www.w3schools.com/react/react_components.asp)

- **WebSite** React: Documentation - Components and Props (https://react.dev/learn#sharing-data-between-components)

- **WebSite** React: Documentation - React.Component (https://react.dev/learn#components)

- **WebSite** MDN Web Docs: Getting started with React - Exploring our first React component (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#exploring_our_first_react_component_%E2%80%94_app)

- **Article** Understanding React Components (https://medium.com/the-andela-way/understanding-react-components-37f841c1f3bb)

- **Article** React component guide: Class vs functional (https://learningdaily.dev/react-component-guide-class-vs-functional-929108fc5afc)

- **YouTube** LearnCode.academy: Reactjs Components & Rendering (https://www.youtube.com/watch?v=fd2Cayhez58)

- **YouTube** Cand Dev: Create Clean and Reusable React Components (https://www.youtube.com/watch?v=lb-80HljuZ4)

- **YouTube** Sonny Sangha: Class Components vs Functional Components in React (https://www.youtube.com/watch?v=yc6elaGOoGQ)

React - Props:

- Props are an object that is injected in components and provides some data that can be shared between other components in an uni-directional flow of data, from a parent to a child element. Props are read-only.

- Passing props

- Manipulating props

Contents

- **WebSite** W3Schools: React Props (https://www.w3schools.com/react/react_props.asp)

- **WebSite** React: Documentation - Components and Props (https://reactjs.org/docs/components-and-props.html)

- **WebSite** React: Documentation - Components and Props (https://reactjs.org/docs/render-props.html#gatsby-focus-wrapper)

- `WebSite` MDN Web Docs: Getting started with React - Variables and props (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#variables_and_props)
- `Article` Props in React (https://medium.com/@wlodarczyk_j/props-in-react-2ec0d0d0df9c)
- `Article` ReactJS Fundamental — Props & State Simplified (https://infinitypaul.medium.com/reactjs-fundamental-props-state-simplified-bb2cd73803d5)
- `YouTube` Sonny Sangha: Learn how to use Props in React in 19 minutes (https://www.youtube.com/watch?v=kHJSNFU7H4U)
- `YouTube` Codevolution: ReactJS Tutorial - 9 - Props (https://www.youtube.com/watch?v=m7OWXtbiXX8)
- `YouTube` The Net Ninja: Full React Tutorial 11 - Props (https://www.youtube.com/watch?v=PHaECbrKgs0)

React Hooks - State:

- The React `useState` Hook allows us to track state in a function component. The `useState` Hook can be used to keep track of strings, numbers, booleans, arrays, objects, etc.
- Control the state of components
- Manipulating variables
- Updating an element or component

Contents

- `WebSite` W3Schools: React Hooks (https://www.w3schools.com/react/react_hooks.asp)
- `WebSite` W3Schools: React useState - Hooks (https://www.w3schools.com/react/react_usestate.asp)
- `WebSite` React: Documentation - Introducing Hooks (https://reactjs.org/docs/hooks-intro.html)
- `WebSite` React: Documentation - Using the State Hook (https://reactjs.org/docs/hooks-state.html)
- `WebSite` MDN Web Docs: State and the useState hook (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_interactivity_events_state#state_and_the_usestate_hook)
- `Article` Learn the useState React Hook (https://adhithiravi.medium.com/learn-the-usestate-react-hook-56511abe4f17)
- `Article` How to Manage State in React with the useState Hook (https://javascript.plainenglish.io/creating-your-second-react-application-d6d50e3e1e91)
- `YouTube` The Net Ninja: Full React Tutorial #8 - Using State (useState hook) (https://www.youtube.com/watch?v=4pO-HcG2igk)
- `YouTube` Sonny Sangha: Learn to use State in React in 19 minutes (for beginners) (https://www.youtube.com/watch?v=kkuq0gTGRFQ)

Creating a React app:

- Create React App is an officially supported way to create single-page React applications. It offers a modern build setup with no configuration.
- Structuring a new React project
- Creating a functional application from scratch

Contents

- `WebSite` Create React App Documentation - Getting Started (https://create-react-app.dev/docs/getting-started)

- `WebSite` MDN Web Docs: Getting started with React - Setting up your first React app (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#setting_up_your_first_react_app)
- `Article` What does Create React App actually do for you? (https://medium.com/rewrite-tech/what-does-create-react-app-actually-do-for-you-c7a9354acdc4)
- `Article` How to set up a new Create React App project (https://medium.com/@dimterion/how-to-set-up-a-new-create-react-app-project-2bfc2992aa16)
- `Article` How to Build a React Project with Create React App in 10 Steps (https://www.freecodecamp.org/news/how-to-build-a-react-project-with-create-react-app-in-10-steps/)
- `YouTube` CodeWithArjun: First React app using create-react-app | VS code | npx | npm (https://www.youtube.com/watch?v=-ERWIp828kY)
- `YouTube` Create React App + Express server (https://www.youtube.com/watch?v=TTZOoWfGjCo)

React Hooks - Effect:

- UseEffect() is a React Hook which allows you to handle side effects in your functional React components.
- Executing a component only after rendering
- Acessing the props of an element
- Calling APIs

Contents

- `WebSite` W3Schools: React Hooks (https://www.w3schools.com/react/react_hooks.asp)
- `WebSite` W3Schools: React useEffect - Hooks (https://www.w3schools.com/react/react_useeffect.asp)
- `WebSite` React: Documentation - Introducing Hooks (https://reactjs.org/docs/hooks-intro.html)
- `WebSite` React: Documentation - Using the Effect Hook (https://reactjs.org/docs/hooks-effect.html)
- `Article` React Hook: A Guide to Learning useEffect() (https://medium.com/swlh/react-hook-a-guide-to-learning-useeffect-11687e777aeb)
- `Article` ReactJS — useEffect() & useCallback() (https://infinitypaul.medium.com/reactjs-useeffect-usecallback-simplified-91e69fb0e7a3)
- `Article` Easy to Learn React Hooks (https://medium.datadriveninvestor.com/easy-to-learn-react-hooks-7db7763608f5)
- `YouTube` The Net Ninja: Full React Tutorial 14 - useEffect Hook (the basics) (https://www.youtube.com/watch?v=gv9ugDJ1ynU)
- `YouTube` Web Dev Simplified: Learn useEffect In 13 Minutes (https://www.youtube.com/watch?v=0ZJgIjIuY7U)

Level 2

React Hooks - Memo:

- The React Hook `useMemo` returns a memoized value. `useMemo` will only recompute the memoized value when one of the dependencies has changed.
- Controlling the state of external variables
- Avoiding expensive calculations on every render

Contents

- `WebSite` W3Schools: React useMemo Hook
  (https://www.w3schools.com/react/react_usememo.asp)

- `WebSite` React: Documentation - useMemo (https://reactjs.org/docs/hooks-reference.html#usememo)

- `WebSite` useHooks.com - useMemo (https://usehooks.com/useMemo/)

- `Article` How to useMemo and useCallback (https://www.developerway.com/posts/how-to-use-memo-use-callback)

- `Article` Use React.memo() wisely (https://dmitripavlutin.com/use-react-memo-wisely/)

- `YouTube` Web Dev Simplified: Learn useMemo In 10 Minutes (https://www.youtube.com/watch?v=THL1OPn72vo)

- `YouTube` Codevolution: React Hooks Tutorial - useMemo Hook (https://www.youtube.com/watch?v=qySZIzZvZOY)

- `YouTube` Dave Gray: Learn useMemo In 10 Minutes (https://www.youtube.com/watch?v=oR8gUi1LfWY)

React Hooks - Callback:

- The React `useCallback` Hook returns a memoized callback function.

- Isolating resource intensive functions so that they will not automatically run on every render

Contents

- `WebSite` W3Schools: React useMemo Hook
  (https://www.w3schools.com/react/react_usecallback.asp)

- `WebSite` React: Documentation - useCallback (https://reactjs.org/docs/hooks-reference.html#usecallback)

- `Article` Your Guide to React.useCallback() (https://dmitripavlutin.com/dont-overuse-react-usecallback/)

- `Article` How to useMemo and useCallback (https://www.developerway.com/posts/how-to-use-memo-use-callback)

- `YouTube` Web Dev Simplified: Learn useCallback In 8 Minutes (https://www.youtube.com/watch?v=_AyFP5s69N4)

- `YouTube` Codevolution: React Hooks Tutorial - useCallback Hook
  (https://www.youtube.com/watch?v=IL82CzlaCys)

React Hooks - Ref:

- The `useRef` Hook allows you to persist values between renders.

- Storing a mutable value that does not cause a re-render when updated

- Accessing a DOM element directly

Contents

- `WebSite` W3Schools: React useRef Hook (https://www.w3schools.com/react/react_useref.asp)

- `WebSite` React: Documentation - useRef (https://reactjs.org/docs/hooks-reference.html#useref)

- `Article` The Complete Guide to useRef() and Refs in React (https://dmitripavlutin.com/react-useref-guide/)

- `Article` React useRef Hook (https://medium.com/trabe/react-useref-hook-b6c9d39e2022)

- `Article` A Thoughtful Way To Use React's useRef() Hook (https://www.smashingmagazine.com/2020/11/react-useref-hook/)
- `YouTube` Web Dev Simplified: Learn useRef in 11 Minutes (https://www.youtube.com/watch?v=t2ypzz6gJm0)
- `YouTube` Codevolution: React Hooks Tutorial - useRef Hook (Part 1) (https://www.youtube.com/watch?v=yCS2m01bQ6w)
- `YouTube` Codevolution: React Hooks Tutorial - useRef Hook (Part 2) (https://www.youtube.com/watch?v=LWg0OyZQffc)

Alura Contents:

- `YouTube` useRef: como funciona esse React Hook (https://www.youtube.com/watch?v=BwRxBGsT_f0)

React - Design System Libraries:

- A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build any number of applications.

Contents

- `WebSite` Awesome React Design Systems (https://github.com/jbranchaud/awesome-react-design-systems#react-design-systems)
- `Article` Top 5 design systems for React (https://bootcamp.uxdesign.cc/top-5-design-systems-for-react-in-2021-643b98f57b0e)
- `Article` The Top React Component Libraries that are Worth Trying (https://technostacks.com/blog/react-component-libraries/)
- `YouTube` TECH IN 5 MINUTES: Ant Design - The best React library? (https://www.youtube.com/watch?v=IEqmSROj5Uc)
- `YouTube` JSWORLD Conference: Sid - What goes into building a design system (https://www.youtube.com/watch?v=KtQO5MhnfEc)
- `YouTube` Anagh Technologies Inc.: Top 5 ReactJS UI Component Libraries (https://www.youtube.com/watch?v=xFXNmA-7VfQ)

React Developer Tools:

- React Developer Tools is used to inspect React components, edit props and state, and identify performance problems.
- Debugging applications

Contents

- `WebSite` Extension React Developer Tools for Google Chrome (https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi)
- `WebSite` React: Documentation - React Developer Tools (https://beta.reactjs.org/learn/react-developer-tools)
- `WebSite` React: Documentation - Profiling Components with the DevTools Profiler (https://reactjs.org/docs/optimizing-performance.html#profiling-components-with-the-devtools-profiler)
- `Article` New React Developer Tools : What's New Features? (https://web-and-mobile-development.medium.com/new-react-developer-tools-whats-new-features-2ae6c1e8a7ea)

- `Article` Debug React applications with React DevTools (https://blog.logrocket.com/debug-react-applications-with-the-new-react-devtools/)

- `YouTube` react.school: React Developer Tools - Installing and Basic Usage (https://www.youtube.com/watch?v=-LfSE9ZxZDI)

- `YouTube` The Net Ninja: Intro to React Dev Tools (https://www.youtube.com/watch?v=rb1GWqCJid4)

Front-end Semantic Versioning:

- SemVer (Semantic Versioning) is a simple set of rules and requirements that dictate how version numbers are assigned and incremented.

- Organizing the dependencies of a project

- Avoiding the "dependency hell"

Contents

- `WebSite` Documentation: Semantic Versioning (https://semver.org/)

- `Article` Semantic versioning for end-user applications (https://uglow.medium.com/making-sense-of-semantic-versioning-for-end-user-software-applications-a3049d97478b)

- `Article` Semantic versioning in npm (https://javascript.plainenglish.io/semantic-versioning-in-npm-54d5b02c3536)

- `Article` Semantic Versioning (https://jwwnz.medium.com/semantic-versioning-f0ac90d7a0fe)

- `YouTube` Swashbuckling with Code: Understanding Semantic Versioning with Real World Examples (https://www.youtube.com/watch?v=1zBzkT7QCmA)

- `YouTube` Steve Griffith - Prof3ssorSt3v3: Semantic Versioning with NPM (https://www.youtube.com/watch?v=mpkC6MmKgsQ)

CSS-in-JS:

- CSS-in-JS refers to a pattern where CSS is composed using JavaScript instead of defined in external files.

- Handling CSS code using JavaScript

- Using styled-components

- Knowing Styled-JSX

Contents

- `WebSite` React: Documentation - Styling and CSS (https://react.dev/learn#adding-styles)

- `WebSite` Microsoft Docs: Style editing for CSS-in-JS frameworks (https://learn.microsoft.com/en-us/microsoft-edge/devtools-guide-chromium/css/css-in-js)

- `WebSite` Chrome Developers: CSS-in-JS support in DevTools (https://developer.chrome.com/blog/css-in-js/)

- `Article` A Thorough Analysis of CSS-in-JS (https://css-tricks.com/a-thorough-analysis-of-css-in-js/)

- `YouTube` Jack Herrington: Introduction to React - CSS in JS (https://www.youtube.com/watch?v=BdzFpyGVfVI)

- `YouTube` Harry Wolff: CSS in JS - My favorite (https://www.youtube.com/watch?v=rKz6cLXhpwA)

- `YouTube` LevelUpTuts: What Is CSS In JS? What Is Styled Components? (https://www.youtube.com/watch?v=EsSi4cER48E)

- `YouTube` Coding Tech: The Past, Present, and Future of CSS-in-JS (https://www.youtube.com/watch?v=a31BUIx-EXc)

Styled Components:

- Styled-components allow you to write actual CSS code to style your components using tagged template literals and the power of CSS.
- Handling CSS code in components

Contents

- `WebSite` Documentation - Styled Components (https://styled-components.com/docs)
- `WebSite` Complete Guide On How To Use Styled-components In React (https://dev.to/elijahtrillionz/complete-guide-on-how-to-use-styled-components-in-react-360c)
- `Article` How To Use Styled-Components In React (https://www.smashingmagazine.com/2020/07/styled-components-react/)
- `YouTube` Traversy Media: Styled Components Crash Course & Project (https://www.youtube.com/watch?v=02zO0hZmwnw)
- `YouTube` PedroTech: Styled Components Full Tutorial - Style Your Components in React (https://www.youtube.com/watch?v=-FZzPHSLauc)

React - Webrouting:

- Manipulating navigation between interfaces and components

Contents

- `WebSite` W3Schools: React Router (https://www.w3schools.com/react/react_router.asp)
- `WebSite` React: Documentation - Route-based code splitting (https://reactjs.org/docs/code-splitting.html#route-based-code-splitting)
- `WebSite` React Router: Documentation (https://reactrouter.com/en/main/start/overview)
- `Article` How To Handle Routing in React Apps with React Router (https://www.digitalocean.com/community/tutorials/how-to-handle-routing-in-react-apps-with-react-router)
- `YouTube` Web Dev Simplified: Learn React Router v6 In 45 Minutes (https://www.youtube.com/watch?v=Ul3y1LXxzdU)
- `YouTube` Web Dev Simplified: How To Create A Navbar In React With Routing (https://www.youtube.com/watch?v=SLfhMt5OUPI)
- `YouTube` The Net Ninja: Full React Tutorial - The React Router (https://www.youtube.com/watch?v=aZGzwEjZrXc)

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)

React Testing Library:

- React Testing Library builds on top of DOM Testing Library by adding APIs for working with React components. It is a set of helpers that let you test React components without relying on their implementation details.

- Testing React components

Contents

- `WebSite` React: Documentation - Testing Overview (https://reactjs.org/docs/testing.html#gatsby-focus-wrapper)

- `WebSite` React Testing Library: Documentação (https://testing-library.com/docs/react-testing-library/intro/)

- `Article` How to build sturdy React apps with TDD and the React Testing Library (https://medium.com/free-code-camp/how-to-build-sturdy-react-apps-with-tdd-and-the-react-testing-library-47ad3c5c8e47)

- `Article` Testing components with Jest and React Testing Library (https://itnext.io/testing-components-with-jest-and-react-testing-library-d36f5262cde2)

- `Article` My experience moving from Enzyme to react-testing-library (https://boyney123.medium.com/my-experience-moving-from-enzyme-to-react-testing-library-5ac65d992ce)

- `YouTube` The Net Ninja: React Testing Library Tutorial (https://www.youtube.com/watch?v=7dTTFW7yACQ&list=PL4cUxeGkcC9gm4_-5UsNmLqMosM-dzuvQ)

- `YouTube` Lama Dev: React Testing Tutorial with React Testing Library and Jest (https://www.youtube.com/watch?v=Flo268xRpV0)

- `YouTube` PedroTech: Testing In React Tutorial - Jest and React Testing Library (https://www.youtube.com/watch?v=JBSUgDxICg8)

Jest:

- Jest is a JavaScript test runner that lets you access the DOM via jsdom. It provides a great iteration speed combined with powerful features like mocking modules and timers so you can have more control over how the code executes.

- Testing components

Contents

- `WebSite` React: Documentation - Testing Overview (https://reactjs.org/docs/testing.html#gatsby-focus-wrapper)

- `WebSite` Jest: Documentation (https://jestjs.io/docs/getting-started)

- `Article` Testing React Components with Jest and Enzyme - In Depth (https://blog.bitsrc.io/how-to-test-react-components-with-jest-and-enzyme-in-depth-145fcd06b90)
- `Article` Testing in React with Jest and Enzyme - An Introduction (https://rossbulat.medium.com/testing-in-react-with-jest-and-enzyme-an-introduction-99ce047dfcf8)
- `Article` Testing components with Jest and React Testing Library (https://itnext.io/testing-components-with-jest-and-react-testing-library-d36f5262cde2)
- `YouTube` Lama Dev: React Testing Tutorial with React Testing Library and Jest (https://www.youtube.com/watch?v=Flo268xRpV0)
- `YouTube` PedroTech: Testing In React Tutorial - Jest and React Testing Library (https://www.youtube.com/watch?v=JBSUgDxlCg8)
- `YouTube` Traversy Media: Jest Crash Course - Unit Testing in JavaScript (https://www.youtube.com/watch?v=7r4xVDI2vho)
- `YouTube` Web Dev Simplified: Introduction To Testing In JavaScript With Jest (https://www.youtube.com/watch?v=FgnxcUQ5vho)

Cypress:

- Cypress is a front-end testing tool that allows for setting up, writing, running, and debugging tests.

Contents

- `WebSite` Cypress: Documentação (https://docs.cypress.io/guides/overview/why-cypress)
- `Article` Why Using Cypress Is Better Than Unit Testing (https://betterprogramming.pub/why-using-cypress-is-better-than-unit-testing-e8234229be81)
- `Article` How to Test Your Frontend with the Cypress.io Framework (https://medium.com/free-code-camp/how-to-test-your-frontend-with-the-cypress-io-framework-f048070f4330)
- `Article` End to End Testing React Apps With Cypress (https://blog.bitsrc.io/testing-react-apps-with-cypress-658bc482678)
- `YouTube` LambdaTest: Learn Cypress in 3 Hours (https://www.youtube.com/watch?v=jX3v3N6oN5M)
- `YouTube` Cypress.io: Cypress in a Nutshell (https://www.youtube.com/watch?v=LcGHiFnBh3Y)
- `YouTube` Fireship: Cypress End-to-End Testing (https://www.youtube.com/watch?v=7N63cMKosIE)

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_manipula%)
- `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)

Babel – Fundamentals:

- Babel is a JavaScript compiler. It is a toolchain that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments.

Contents

- `WebSite` Documentation: What is Babel? (https://babeljs.io/docs/en/)
- `Article` Tutorial - Babel (https://medium.com/@dustinhuang21/tutorial-babel-aaf3162797de)
- `Article` What are NPM, Yarn, Babel, and Webpack; and how to properly use them? (https://medium.com/front-end-weekly/what-are-npm-yarn-babel-and-webpack-and-how-to-properly-use-them-d835a758f987)
- `Article` A brief introduction to Babel.js (https://medium.com/jspoint/how-to-quickly-transpile-javascript-using-babel-alone-a-brief-introduction-to-babel-js-40e74e43fe32)
- `YouTube` codedamn: What is Babel and why you need it? Introduction to Babel 7 (https://www.youtube.com/watch?v=yLrNwo4wXOs)
- `YouTube` Strezabyte: Babel.js - What It Is, and How You Can Use It (https://www.youtube.com/watch?v=C2PDAGCrk_g)
- `YouTube` Monsterlessons Academy: Getting Started With Babel - Transpiling Javascript (https://www.youtube.com/watch?v=o9hmjdmJLMU)
- `YouTube` Daily Tuition: What is Use of Babel in React - React For Beginners (https://www.youtube.com/watch?v=PUoFXd3b5CY)

Level 3

Lottie:

- Lottie is a library that parses Adobe After Effects animations exported as json with Bodymovin and renders them natively on mobile and on the web.

Contents

- `WebSite` Documentation: Lottie (https://airbnb.io/lottie/#/README)
- `WebSite` Lottie Components (https://lottiereact.com/components/Lottie)

- `Article` A Guide to Lottie Framework: 5 Steps to Create an Animation (https://medium.muz.li/a-guide-to-lottie-framework-5-steps-to-create-an-animation-6ccc446d7b13)
- `Article` Native Animation For Mobile App Using Lottie (https://blog.prototypr.io/native-animation-for-mobile-app-using-lottie-eafe0640ece4)
- `Article` Lottie React Native Tutorial (https://medium.com/react-native-training/lottie-react-native-tutorial-162d91840720)
- `Article` Lottie Without After Effects (https://medium.com/haiku-blog/lottie-without-after-effects-9c5a8e74c239)
- `YouTube` LottieFiles: What is a Lottie animation? (https://www.youtube.com/watch?v=fVl60ZyO_MA)
- `YouTube` Design Pilot: Create Lottie Animations in After Effects - The Ultimate Guide (https://www.youtube.com/watch?v=72qbebvwxnY)
- `YouTube` LottieFiles: How to animate SVG icon and convert to Lottie: SVG to Lottie converter (https://www.youtube.com/watch?v=uT8p_0VLxTs)

Framer Motion:

- Framer Motion is a production-ready motion library for React from Framer.
- Creating animations

Contents

- `WebSite` Documentation: Framer Motion (https://www.framer.com/docs/)
- `Article` Introduction to Framer Motion for React (https://blog.bitsrc.io/introduction-to-framer-motion-for-react-4ba5a9cabc1b)
- `Article` Framer Motion Tutorials: Make More Advanced Animations (https://betterprogramming.pub/framer-motion-tutorials-make-more-advanced-animations-4344b686ea0a)
- `Article` Create Stunning Animations In React Using Framer Motion (https://betterprogramming.pub/create-amazing-animations-in-react-using-framer-motion-34c803f60c6f)
- `Article` Smooth Animations With React and Framer Motion (https://betterprogramming.pub/smooth-animations-with-react-and-framer-motion-c272b6f22f67)
- `YouTube` PedroTech: Animations In React - Framer-Motion Tutorial (https://www.youtube.com/watch?v=GOuwOI-WSkE)
- `YouTube` Fireship: Framer Motion & React Tutorial for Beginners (https://www.youtube.com/watch?v=SuqU904ZHA4)
- `YouTube` Laith Academy: Framer Motion (React Animation Library) Crash Course (https://www.youtube.com/watch?v=1vKiPwEYbyk)

Service Workers:

- Service workers essentially act as proxy servers that sit between web applications, the browser, and the network (when available). They are intended, among other things, to enable the creation of effective offline experiences, intercept network requests and take appropriate action based on whether the network is available, and update assets residing on the server. They will also allow access to push notifications and background sync APIs.

Contents

- `WebSite` MDN Web Docs: Service Worker API (https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API)

- `WebSite` MDN Web Docs: Using Service Workers (https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers)

- `WebSite` Chrome Developers: Service worker overview (https://developer.chrome.com/docs/workbox/service-worker-overview/)

- `Article` Service workers: the little heroes behind Progressive Web Apps (https://medium.com/free-code-camp/service-workers-the-little-heroes-behind-progressive-web-apps-431cc22d0f16)

- `Article` Understanding Service Workers and Caching Strategies (https://blog.bitsrc.io/understanding-service-workers-and-caching-strategies-a6c1e1cbde03)

- `Article` An Introduction to Service Workers in JavaScript (https://codeburst.io/an-introduction-to-service-workers-in-javascript-27d6376460c2)

- `Article` Service Workers in React (https://levelup.gitconnected.com/service-workers-in-react-8bcaccfd0356)

- `Article` Beginners guide to Web Push Notifications using Service Workers (https://medium.com/@a7ul/beginners-guide-to-web-push-notifications-using-service-workers-cb3474a17679)

- `Article` PWA: Create a `New Update Available` Notification using Service Workers (https://medium.com/progressive-web-apps/pwa-create-a-new-update-available-notification-using-service-workers-18be9168d717)

- `YouTube` Google Chrome Developers: Introduction to Service Workers (https://www.youtube.com/watch?v=jVfXiv03y5c)

- `YouTube` Google Chrome Developers: Introduction to Service Worker - Progressive Web App Training (https://www.youtube.com/watch?v=wEPeaJgbIxQ)

- `YouTube` Traversy Media: Intro To Service Workers & Caching (https://www.youtube.com/watch?v=ksXwaWHCW6k)

- `YouTube` The Net Ninja: PWA Tutorial for Beginners - Intro to Service Workers (https://www.youtube.com/watch?v=hxiggHZOGlQ)

React Hooks - Form:

- React Hooks Form is a library that provides form validation.

Contents

- `WebSite` React Hook Form: Documentação (https://react-hook-form.com/get-started)

- `Article` React Hook Form VS Formik (https://blog.bitsrc.io/react-hook-form-vs-formik-form-builder-library-for-react-23ed559fdae)

- `Article` React Hook Form — An Elegant Solution to Forms in React (https://blog.bitsrc.io/react-hook-form-an-elegant-solution-to-forms-in-react-4db64505b0cd)

- `Article` How to Use React Hook Form with TypeScript (https://javascript.plainenglish.io/how-to-use-react-hook-form-with-typescript-2cf597c0c45f)

- `Article` How to Use the React Hook Form With Ionic React Components (https://betterprogramming.pub/how-to-use-react-hook-form-with-ionic-react-components-eaa4426d8a2d)

- `YouTube` Maksim Ivanov: React Hook Form Tutorial - How To Create Forms In React (https://www.youtube.com/watch?v=bU_eq8qyjic)

- Beier Luo: React Hook Form - Get Started (https://www.youtube.com/watch?v=RkXv4AXXC_4)
- `YouTube` PedroTech: React Forms Full Tutorial - Validation, React-Hook-Form, Yup (https://www.youtube.com/watch?v=UvH70UkbyfE)

Lodash:

- Lodash is a modern JavaScript utility library delivering modularity, performance & extras. Lodash's modular methods are great for iterating arrays, objects, & strings; manipulating & testing values; and creating composite functions.

Contents

- `WebSite` Documentation: Lodash (https://lodash.com/)
- `Article` Introduction to Lodash (https://medium.com/front-end-weekly/introduction-to-lodash-71dbee093b49)
- `Article` 10 Lodash functions everyone should know (https://medium.com/voobans-tech-stories/10-lodash-functions-everyone-should-know-334b372aec5d)
- `YouTube` iEatWebsites: What is Lodash and How it Works (for beginners) (https://www.youtube.com/watch?v=YyxliogSwrM)
- `YouTube` FireJet: 10 things you should know about Lodash (https://www.youtube.com/watch?v=2JKGGadhFxU)
- `YouTube` Mahmud Ahsan: JavaScript Lodash Tutorial (https://www.youtube.com/watch?v=OOyWZnZrl7M)

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

- `WebSite` GraphQL: Documentation (https://graphql.org/)
- `WebSite` How to GraphQL: The Fullstack tutorial for GraphQL (https://www.howtographql.com/)
- `WebSite` RedHat: 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)

Redux Saga:

- Redux Saga is a library that aims to make application side effects (i.e. asynchronous things like data fetching and impure things like accessing the browser cache) easier to manage, more efficient to execute, easy to test, and better at handling failures.

Contents

- **WebSite** Documentation: Redux Saga (https://redux-saga.js.org/docs/introduction/GettingStarted)
- **Article** What is Redux-Saga? (https://www.javatpoint.com/what-is-redux-saga)
- **Article** Understanding Redux Saga: From action creators to sagas (https://blog.logrocket.com/understanding-redux-saga-action-creators-sagas/)
- **Article** A high level introduction to Redux Saga (https://www.loginradius.com/blog/engineering/introduction-to-redux-saga/)
- **Article** React Redux Saga example app (https://medium.com/@lavitron/make-your-first-call-to-api-using-redux-saga-15aa995df5b6)
- **Article** Redux-Saga tutorial for beginners and dog lovers (https://medium.com/hackernoon/redux-saga-tutorial-for-beginners-and-dog-lovers-aa69a17db645)
- **YouTube** ReactJS Hub: Introduction to Redux Saga and getting data from APIs (https://www.youtube.com/watch?v=1K26DIKt3w8)
- **YouTube** EdRoh: Understanding Redux Saga with a simple API call Tutorial (https://www.youtube.com/watch?v=eA6rskkE9y8)
- **YouTube** Duomly: How to implement redux saga with ReactJS and Redux - tutorial (https://www.youtube.com/watch?v=1EVwGxXU84w)

NextJS - Fundamentals:

- Next.js is an open-source web development framework created by Vercel enabling React-based web applications with server-side rendering and generating static websites.
- Building web interfaces
- Decreasing page load times
- Rendering server-side pages
- Improving performance in React
- Building API routes with serveless functions
- CSS-in-JS

Contents

- `WebSite` Documentation: NextJS (https://nextjs.org/docs)
- `Article` Build Better React apps with Next.js - Next.js tutorial with examples (https://javascript.plainenglish.io/build-better-react-apps-with-next-js-d7bc7b495006)
- `Article` Build A Chat App With Sentiment Analysis Using Next.js (https://codeburst.io/build-a-chat-app-with-sentiment-analysis-using-next-js-c43ebf3ea643)
- `Article` Create Next.js Application and Deploy to Vercel (https://okandavut.medium.com/create-next-js-application-and-deploy-to-vercel-4210c51b071f)
- `Article` Next.js Authentication Tutorial (https://medium.com/@auth0/next-js-authentication-tutorial-99111927a55)
- `YouTube` Academind: Next.js Crash Course for Beginners (https://www.youtube.com/watch?v=MFuwkrseXVE)
- `YouTube` Traversy Media: Next.js Crash Course (https://www.youtube.com/watch?v=mTz0GXj8NN0)
- `YouTube` Fireship: Next.js in 100 Seconds - Plus Full Beginner's Tutorial (https://www.youtube.com/watch?v=Sklc_fQBmcs)

Auxiliary Skill: Infrastructure and Back-end

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)

JSON:

- JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.
- Creating an object
- Transforming an object into a string
- Transforming a string into an object
- Manipulating an object

Contents

- `WebSite` W3Schools: JSON (https://www.w3schools.com/js/js_json_intro.asp)
- `WebSite` MDN Web Docs: JSON (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- `WebSite` MDN Web Docs: Working with JSON (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON)
- `YouTube` Web Dev Simplified: Learn JSON in 10 Minutes (https://www.youtube.com/watch?v=iiADhChRriM)
- `YouTube` Traversy Media: JSON Crash Course (https://www.youtube.com/watch?v=wI1CWzNtE-M)

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/)

YARN:

- Yarn is a package manager for your code. It allows you to use and share code with other developers. Code is shared through something called a package (sometimes referred to as a module). A package contains all the code being shared as well as a package.json file which describes the package.
- Managing packages
- Managing dependencies
- Installing packages offline
- Commands
- The yarn.lock file

Contents

- `WebSite` YARN: Documentation (https://classic.yarnpkg.com/en/docs)
- `Article` NPM vs Yarn Cheat Sheet (https://shift.infinite.red/npm-vs-yarn-cheat-sheet-8755b092e5cc)
- `Article` What are NPM, Yarn, Babel, and Webpack; and how to properly use them? (https://medium.com/front-end-weekly/what-are-npm-yarn-babel-and-webpack-and-how-to-properly-use-them-d835a758f987)
- `Article` npm vs Yarn — Choosing the right package manager (https://javascript.plainenglish.io/npm-vs-yarn-choosing-the-right-package-manager-a5f04256a93f)
- `YouTube` Traversy Media: Yarn Package Manager Crash Course (https://www.youtube.com/watch?v=g9_6KmiBISk)
- `YouTube` Clever Programmer: NPM vs Yarn - Which is the best Package Manager? (https://www.youtube.com/watch?v=0DGCIZD5LEM)

Auxiliary Skill: UX and Design

Design Systems:

- A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build applications.
- Creating and maintaining libraries that will be consumed and used as a standard for building a project

Contents

- `Article` Everything you need to know about Design Systems (https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969)
- `Article` A comprehensive guide to design systems (https://www.invisionapp.com/inside-design/guide-to-design-systems/)
- `Article` What is a UX Design System? (https://fuzzymath.com/blog/what-is-ux-design-system/)
- `Article` Creating a design system in Figma — Tutorial 1 (https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb)
- `YouTube` DesignerUp: 5 Best Design Systems and How to Learn (and Steal) From Them (https://www.youtube.com/watch?v=BISC15OPeGA)
- `YouTube` DesignCourse: What is a Design System? Design Systems 101 for Designers (https://www.youtube.com/watch?v=wc5krC28ynQ)
- `YouTube` vaexperience: Design Systems - What are They and How to Get Started (https://www.youtube.com/watch?v=I6YuGE6EjJA)

Figma - Fundamentals:

- Figma is a collaborative web application for interface design. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilising a variety of vector graphics editor and prototyping tools.
- Creating page layouts and components

Contents

- `Article` Getting Started with Figma (https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017)
- `Article` Introducing Figma to React (https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc)
- `YouTube` Adrian Twarog: Figma Crash Course (https://www.youtube.com/watch?v=Ig7w3Ntfqy0)
- `YouTube` AJ&Smart: Figma UI Design Tutorial - Get Started in Just 24 Minutes (https://www.youtube.com/watch?v=FTFaQWZBqQ8)
- `YouTube` Jesse Showalter: Intro to Figma - Beginners guide to Figma Basics (https://www.youtube.com/watch?v=jk1T0CdLxwU)
- `YouTube` Figma: Figma Tutorial - Components - The Basics (https://www.youtube.com/watch?v=k74IrUNaJVk)
- `YouTube` Figma: Figma Components 101 (https://www.youtube.com/watch?v=jk1T0CdLxwU)
- `Course` LearnUX: Figma Course (https://learnux.io/course/figma)

Design components:

- Knowing the components that describe a layout or interface

Contents

- `Article` Text fields & Forms design — UI components series (https://uxdesign.cc/text-fields-forms-design-ui-components-series-2b32b2beebd0)

- `Article` Design Components for Emails (https://blog.prototypr.io/design-components-for-emails-587f6267951d)

- `YouTube` GCFLearnFree.org: Beginning Graphic Design - Layout & Composition (https://www.youtube.com/watch?v=a5KYlHNKQB8)

- `YouTube` uxable - learning ux together: UI Elements/Components - Types and Importance of UI Elements (https://www.youtube.com/watch?v=roF6ocfArK0)

- `YouTube` Dennis Cortes: How to Design Powerful Components and Buttons with Variants and Auto Layout in Figma (https://www.youtube.com/watch?v=ODOR5EniJNg)

- `YouTube` Tutorial Tim: Let's Build a Design System - Understanding Layout (https://www.youtube.com/watch?v=nn3tO7gAFR4)

Color systems:

- Defining a color palette that makes sense for a given interface

Contents

- `Article` Color Theory: There is more than one set of universal Primary Colors (https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9)

- `Article` The Science of Color (https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea)

- `Article` RGB vs. CMYK: A guide to color systems for designers (https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554)

- `Article` Color in UI Design: A (Practical) Framework (https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e)

- `Article` Basic UI color guide (https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a)

- `YouTube` Rachel How: How to pick the right colors for your website or app (https://www.youtube.com/watch?v=ewRYw4pnKQU)

- `YouTube` Flux Academy: How to Choose Colors (Easy 3-Step Process) (https://www.youtube.com/watch?v=KMS3VwGh3HY)

- `YouTube` Jesse Showalter: 60-30-10 Color Rule (https://www.youtube.com/watch?v=UWwNIMHFdW4)

- `YouTube` DesignerUp: Super Practical Guide to Color Theory, Color Models and Perfect Color Palettes (https://www.youtube.com/watch?v=GyVMoejbGFg)

How to use Fonts:

- Choosing the most appropriate font for a given project

Contents

- `Article` Top 10 UI Fonts for Web & Mobile (https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3)

- `Article` Best Fonts for UI Design, I use Daily - Best Typefaces and Font Resources for UI Designers (https://uxplanet.org/best-fonts-for-ui-design-i-use-daily-4a7bcffb966c)

- `YouTube` DesignerUp: Best Practices for Choosing Fonts and Font Pairing in UI and Web Design (https://www.youtube.com/watch?v=30IoKUGaPWQ)

- YouTube DesignerUp: The Difference Between Fonts, Typefaces and Typography for UI Designers (https://www.youtube.com/watch?v=OKGTkLqgm58)

- YouTube DesignerUp: Choosing and Pairing Fonts - UI Design (https://www.youtube.com/watch?v=uWCWTq1cPW0)

- YouTube Mizko: The ONLY 8 Fonts UI Designers Need. Forget The Rest. (https://www.youtube.com/watch?v=mEAmAFgzQd4)