**Introduction To GIT**

** What Is** [**Version Control**](https://www.atlassian.com/git/tutorials/what-is-version-control)

**** [**What is Git**](https://www.atlassian.com/git/tutorials/what-is-git)

**** [**Git Basics**](https://www.atlassian.com/git)

** Setting Up** [**Repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository)

**** [**Merging vs Rebasing**](https://www.atlassian.com/git/tutorials/merging-vs-rebasing)

**** [**Resetting, Checking-out, and Reverting**](https://www.atlassian.com/git/tutorials/resetting-checking-out-and-reverting)

**Advanced Tips And Concepts**

**** [**Branching Strategy**](https://medium.com/@sreekanth.thummala/choosing-the-right-git-branching-strategy-a-comparative-analysis-f5e635443423) **/** [**Branching Strategies**](https://tilburgsciencehub.com/topics/automation/version-control/advanced-git/git-branching-strategies/)

**** [**Git Log**](https://www.atlassian.com/git/tutorials/git-log)

**** [**Cherry Pick**](https://www.atlassian.com/git/tutorials/cherry-pick)

**** [**About Bash**](https://www.atlassian.com/git/tutorials/git-bash)

**CI/CD**

* [CI/CD Basic](https://github.com/resources/articles/devops/ci-cd) (What, Why and How)

CI/CD Tools

* [Github Actions](https://docs.github.com/en/actions)
* [Jenkins](https://www.jenkins.io/doc/)

**Introduction To HTML**

* [Introduction to HTML](https://ebook.learn.epam.com/html-course/docs/category/1-introduction-to-html)
* [Document Structure](https://web.dev/learn/html/document-structure)

**Basic Of HTML**

**** [**HTML Basic**](https://www.w3schools.com/html/html_basic.asp)

* **html**
* **head**
* **body**
* **heading**
* **paragraph**
* **links**
* **images**

**** [**HTML Elements**](https://www.w3schools.com/html/html_elements.asp)

* **Nested Elements**
* **Tags**

**** [**HTML Attributes**](https://www.w3schools.com/html/html_attributes.asp)

* **src**
* **lang**
* **href**
* **title**
* **width, height**
* **alt**
* **style**
* **class**
* **id**

**** [**Text Formatting**](https://www.w3schools.com/html/html_formatting.asp)

**** [**Block And Inline Element**](https://www.w3schools.com/html/html_blocks.asp)

**** [**HTML Layout**](https://www.w3schools.com/html/html_layout.asp)

**** [**HTML Behind The Scenes**](https://blog.logrocket.com/how-browser-rendering-works-behind-scenes/)

**** [**Web Accessibility**](https://developer.mozilla.org/en-US/docs/Web/Accessibility)

**Semantic HTML**

[**Semantic Elements**](https://www.w3schools.com/html/html5_semantic_elements.asp)

[**Semantic Elements Introduction**](https://ebook.learn.epam.com/html-course/docs/semantic-markup/introduction)

* **Header**
* **Nav**
* **Main**
* **Aside**
* **Article**
* **Footer**

**Advanced HTML**

** List (ul, ol, li)**

**** [**HTML Table Basics**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Basics)

** Forms**

** Integration of Css and JS file**

**** [**What is Web APIs**](https://www.w3schools.com/js/js_api_intro.asp)

**** [**Introduction To Web APIs**](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction)

* [**Web Storage APIs**](https://www.w3schools.com/js/js_api_web_storage.asp)
  + **localStorage**
  + **sessionStorage**

**** [**What Is Web Workers**](https://www.w3schools.com/js/js_api_web_workers.asp)

* **Create Web Worker**

**** [**Web Geolocation API**](https://www.w3schools.com/js/js_api_geolocation.asp)

** HTML API (GeoLocation, Web Storage, Web Workers , Server Sent Event)**

**** [**HTML Graphics**](https://www.w3schools.com/graphics/)

* **Canvas**
* **SVG**

**** [**HTML Media**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Video_and_audio_content)

* **Audio**
* **Video**

**** [**Debugging HTML**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Debugging_HTML)

**Introduction To CSS**

[**Introduction to CSS**](https://ebook.learn.epam.com/css-course/docs/category/1-introduction-to-css)

* [**What is CSS**](https://ebook.learn.epam.com/css-course/docs/Intro/about)
* [**How CSS Works With HTML**](https://ebook.learn.epam.com/css-course/docs/Intro/how_it_works)
* **Ways to add CSS to HTML**
  + [**Inline External and Internal**](https://ebook.learn.epam.com/css-course/docs/Intro/inline_external_internal)
* [**CSS Syntax**](https://www.w3schools.com/css/css_syntax.asp)

**Basic Of CSS**

**** [**Box Model**](https://www.w3schools.com/css/css_boxmodel.asp)

* **padding**
* **margin**
* **border**
* **content**

**** [**Outline**](https://www.w3schools.com/css/css_syntax.asp)

**** [**CSS With Table**](https://www.w3schools.com/css/css_table.asp)

**** [**CSS Links**](https://www.w3schools.com/css/css_link.asp)

**** [**CSS Display**](https://www.w3schools.com/css/css_display_visibility.asp)

* **inline**
* **block**
* **none**

**** [**CSS Fonts**](https://www.w3schools.com/css/css_font.asp)

**** [**Selectors**](https://www.w3schools.com/css/css_selectors.asp) **And** [**Combinators**](https://www.w3schools.com/css/css_combinators.asp)

**** [**Opacity**](https://www.w3schools.com/css/css_image_transparency.asp)

**** [**Height ,Width And Max-width**](https://www.w3schools.com/css/css_dimension.asp)

**CSS Preprocessor, Flexbox And Grid**

**** [**What Is Preprocessor**](https://developer.mozilla.org/en-US/docs/Glossary/CSS_preprocessor)

**** [**Sass Preprocessor**](https://sass-lang.com/) **/** [**Less Preprocessor**](https://lesscss.org/) **(Any One )**

**** [**Which One To Choose**](https://www.frontendmentor.io/articles/css-preprocessors-sass-or-less-which-to-choose-JOI20I1xNL)

**** [**Flow Layout**](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flow_layout/Block_and_inline_layout_in_normal_flow)

**** [**Basic of Flexbox**](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout/Basic_concepts_of_flexbox) **/**[**Basic of Grid**](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout) **Layout**

* **Layout**
* **Containers**
* **Items**
* **responsive**
* **Reverse Order/flow**

** Playground for Flexbox and Grid**

* [**https://flexboxfroggy.com/**](https://flexboxfroggy.com/)
* [**https://cssgridgarden.com**](https://cssgridgarden.com/)**/**

**Advanced CSS**

**** [**CSS Position**](https://www.w3schools.com/css/css_positioning.asp)

* **static**
* **Relative**
* **absolute**
* **fixed**
* **sticky**

**** [**Z-Index**](https://www.w3schools.com/css/css_z-index.asp)

**** [**What is Responsive Design**](https://www.w3schools.com/css/css_rwd_intro.asp)**/** [**About Responsive Design**](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design)

**** [**Responsive Design With Media Queries**](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries)

**** [**Pseudo Classes**](https://www.w3schools.com/css/css_pseudo_classes.asp)

**** [**Pseudo Elements**](https://www.w3schools.com/css/css_pseudo_elements.asp)

** CSS Frameworks (Any one)**

* [**CSS With Bootstrap**](https://getbootstrap.com/docs/3.4/css/)
* [**CSS With Tailwind**](https://tailwindcss.com/docs/installation)

**** [**CSS Methodologies**](https://medium.com/sahibinden-technology/using-css-methodologies-b61c3a6c6c9f)

**** [**CSS Transitions**](https://www.w3schools.com/css/css3_transitions.asp)

**Javascript Basics**

**﻿**[**﻿**](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics)**Reference Link :** [**JavaScript Basics**](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics)**﻿**

* **Javascript where to use**
* **variables (var, let, const)**
* **Data Types**
* **Operators**
* **Browser console**

**Control Structure**

**Conditionals:**[**Control Strucuture**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)[**﻿**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)**﻿**

* **If**
* **Else**
* **Switch**

**﻿**[**﻿**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)**﻿**[**Loops**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration)**﻿**

* **for**
* **while**
* **do-while**
* **for...of**
* **for...in**

**Functions**

**Reference Link -** [**﻿**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions)**﻿**[**Functions**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions)**﻿**

* **Function Declarations**
* **Function Expressions**
* **Function Hoisting**
* **Function Scope**
* **Arrow Functions**
* **Higher Order Function**
* **Function Stack**
* **Nested Functions And Closures**
* **Rest Parameters**
* **closure and scoping**
  + **global scope**
  + **local scope**
  + **lexical scoping**
  + **Dynamic scoping**

**Functional Programming**

**Reference Link -** [**Functional Programming in JS**](https://www.freecodecamp.org/news/functional-programming-in-javascript/)**﻿**

* **Immutability and side effects**
* **Pure Function**
* **Currying**
* **Memoization**
* **function Composition**
* **Lodash**

**DOM Manipulation And Storage**

**﻿**[**https://developer.mozilla.org/en-US/docs/Web/API/Document\_Object\_Model/Introduction**](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)**﻿**

* **Selecting and Modifying DOM elements**
* **Handling events**
* **use of browser storage** 
  + **local storage**
  + **session storage**
  + **cookies**

**Object Oriented Programming (OOP) In JS**

**Reference Link-** [**Working with Objects**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects)**﻿**

* **Object Properties And Methods**
* **Object creation patterns**
* **Object literals**
* **Object.create**
* **Traverse to object property using loop**
* **Add or Delete new property of object**
* **Comparing Objects**
* **﻿**[**Inheritance and Prototype**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain)**﻿**
* **Constructor function and new keyword**
* **Object clone (**[**Deep Copy**](https://developer.mozilla.org/en-US/docs/Glossary/Deep_copy) **,** [**Shallow Copy**](https://developer.mozilla.org/en-US/docs/Glossary/Shallow_copy)**)**
* **﻿**[**Understanding of This**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/this)**﻿**
  + **call , bind, apply**
  + **this in different context ( global, function, method)**

**Arrays And Iterables**

**Reference Link -** [**Arrays**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)**﻿**

* **Array Methods** 
  + **map, filter, forEach, reduce**
  + **flat, flatMap**
  + **concat, slice, splice, reverse, lastIndexOf, indexOf**
* **﻿**[**Spread**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax) **operators**
* **﻿**[**Rest**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/rest_parameters) **parameters**

**Advance Javascript**

* **ES6 and Beyond(** [**ES6 Features**](https://www.w3schools.com/js/js_es6.asp) **)**
  + **ES6 features (const, let, template literals, destructuring )**
  + **new data structure (Set, Map)**
  + **classes**
  + **modules (import, export)**
  + **﻿**[**Asynchronous JavaScript**](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous)**﻿**
    - **Promises**
    - **async/ await** [**Event Loop**](https://www.youtube.com/watch?v=8zKuNo4ay8E)**﻿**
    - **callbacks**
    - **Error Handling (try , catch)**
* **HTTP Requests and response**
  + **fetch API**
  + **XMLHttpRequest**
  + **﻿**[**HTTP Payload Body**](https://developer.mozilla.org/en-US/docs/Glossary/Payload_body)
  + **﻿**[**HTTP Request Methods**](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods)**(GET, POST, PUT, DELETE)**
  + **﻿**[**HTTP Response Status Code**](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status)**﻿**

**Testing , Debugging, Optimization And Security**

* **﻿**[**Unit Testing with Jest**](https://jestjs.io/)**﻿**
* **﻿**[**Debugging JS**](https://developer.chrome.com/docs/devtools/javascript)**﻿**
* **﻿**[**Performance Optimization**](https://developer.mozilla.org/en-US/docs/Learn/Performance/JavaScript)**﻿**
* **﻿**[**Security**](https://developer.mozilla.org/en-US/docs/Web/Security)**﻿**
* **﻿**[**OWASP Top 10 Security Risks**](https://owasp.org/www-project-top-ten/)

**Introduction To TypeScript**

* Why TypeScript ?
* Setting up TypeScript
* What is tsconfig.json ([TS Config JSON](https://www.typescriptlang.org/docs/handbook/tsconfig-json.html))
* Basic types ([Basic Types](https://www.typescriptlang.org/docs/handbook/2/everyday-types.html))
* Variables
* Functions ([With Functions](https://www.typescriptlang.org/docs/handbook/2/functions.html))
* Classes ([With Classes](https://www.typescriptlang.org/docs/handbook/2/classes.html))
* Interfaces
* types vs Interfaces
* Public Private Modifiers
* Method Decorators

﻿[Cheatsheets Reference](https://www.typescriptlang.org/cheatsheets/" \t "_blank)﻿

**Advanced Types And Modules**

Reference Link - [Advanced Types](https://www.typescriptlang.org/docs/handbook/advanced-types.html)﻿

* Generics
* Enums
* union
* Intersection
* Type guards
* Narrowing
* Working with Modules ([﻿](https://www.typescriptlang.org/docs/handbook/modules/theory.html)﻿[Modules](https://www.typescriptlang.org/docs/handbook/modules/theory.html))
  + Modules and Namespace
  + Import, Export
  + Default and wildcard usage

**Integration With Tools And Frameworks**

* Configure TypeScript with webpack ([Integration With Build Tools](https://www.typescriptlang.org/docs/handbook/integrating-with-build-tools.html#webpack))
* Typescript with Jest for testing
* TypeScript with React, Angular
* Typescript with NodeJS
* ﻿[Project Reference](https://www.typescriptlang.org/docs/handbook/project-references.html)

**Tools And Best Practices**

**Best Practices Guide**

* Clean Code [Clean Code Book](https://github.com/jnguyen095/clean-code/blob/master/Clean.Code.A.Handbook.of.Agile.Software.Craftsmanship.pdf)﻿
* ﻿[SOLID Principles](https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/), [DRY, KISS and YAGNI](https://www.linkedin.com/pulse/best-practices-writing-javascript-code-dry-kiss-yagni-emmanuel/) / [Principles of Software Engineering and Clean Code](https://medium.com/@hlfdev/kiss-dry-solid-yagni-a-simple-guide-to-some-principles-of-software-engineering-and-clean-code-05e60233c79f)﻿

**Static Code Analyser**

* Sonarqube -> [What is Sonarqube](https://docs.sonarsource.com/sonarqube-server/latest/)﻿
* ﻿[EsLint](https://eslint.org/docs/latest/use/getting-started" \t "_blank)﻿

﻿

**VS Code Extension**

* ﻿[Prettier](https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode)﻿
* ﻿[Git Lens](https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens)﻿
* ﻿[Live Server](https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer)﻿
* ﻿[Github Copilot](https://code.visualstudio.com/docs/copilot/overview" \t "_blank)﻿
* ﻿[Material Icon Theme](https://marketplace.visualstudio.com/items?itemName=PKief.material-icon-theme)﻿

**Chrome Plugins**

* ﻿[React Developer Tools](https://chromewebstore.google.com/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en)﻿
* ﻿[Redux Dev Tools](https://chromewebwebstore.google.com/detail/redux-devtools/lmhkpmbekcpmknklioeibfkpmmfibljd?hl=en)﻿
* ﻿[Google Lighthouse](https://developer.chrome.com/docs/lighthouse/overview)﻿

**Advanced Plugins**

* Memory Snapshot [Heap Snapshots](https://developer.chrome.com/docs/devtools/memory-problems/heap-snapshots)﻿
* ﻿[Chrome Devtools](https://developer.chrome.com/docs/devtools/overview/)﻿

﻿

**Introduction to React**

* ﻿[What Is React](https://legacy.reactjs.org/tutorial/tutorial.html#what-is-react)﻿
* ﻿[Thinking In React](https://react.dev/learn/thinking-in-react)﻿
* Write component [With JSX](https://react.dev/learn/writing-markup-with-jsx)
* ﻿[Component](https://react.dev/reference/react/Component)﻿
* ﻿[Props](https://react.dev/learn/passing-props-to-a-component#familiar-props) And [State](https://react.dev/learn/passing-props-to-a-component#familiar-props) ﻿
* ﻿[React Lifecycle In Class](https://legacy.reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-a-class)
* ﻿[Lifecycle Of Effect](https://react.dev/learn/lifecycle-of-reactive-effects#the-lifecycle-of-an-effect)﻿
* How class Lifecyclce works with Hooks
* ﻿[Conditional Rendering](https://react.dev/learn#conditional-rendering)﻿
* ﻿[SSR in JS](https://www.freecodecamp.org/news/server-side-rendering-javascript/)﻿
* ﻿[SSR vs CSR](https://ritikchourasiya.medium.com/understanding-the-difference-between-csr-and-ssr-rendering-d2fa0eee50e)﻿

**Core Concepts**

* ﻿[Virtual DOM](https://legacy.reactjs.org/docs/faq-internals.html#what-is-the-virtual-dom) (List, Key and Reconciliation)
* ﻿[Lifecycle](https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/)﻿
* ﻿[Component Composition](https://legacy.reactjs.org/docs/composition-vs-inheritance.html#gatsby-focus-wrapper)﻿
* ﻿[Lifting State](https://react.dev/learn/sharing-state-between-components#lifting-state-up-by-example)﻿
* ﻿[Controlled And Uncontrolled Components](https://react.dev/learn/sharing-state-between-components#controlled-and-uncontrolled-components) / React Forms
* ﻿[HOC](https://legacy.reactjs.org/docs/higher-order-components.html#gatsby-focus-wrapper)﻿
* ﻿[Dynamic Styles](https://react.dev/reference/react/useInsertionEffect#injecting-dynamic-styles-from-css-in-js-libraries)﻿
* ﻿[Fragment](https://react.dev/reference/react/Fragment)﻿
* ﻿[Profiler](https://react.dev/reference/react/Profiler)﻿
* ﻿[React Lazy](https://react.dev/reference/react/lazy)﻿
* ﻿[Code Splitting](https://legacy.reactjs.org/docs/code-splitting.html#code-splitting)﻿
* ﻿[Fallback UI](https://react.dev/reference/react/Suspense) / [Error Boundary / Where To Place Error Boundary](https://react.dev/reference/react/useTransition#displaying-an-error-to-users-with-error-boundary)﻿
* ﻿[React Devtools](https://react.dev/learn/tutorial-tic-tac-toe#react-developer-tools)﻿
* ﻿[Accessibility](https://legacy.reactjs.org/docs/accessibility.html#gatsby-focus-wrapper)﻿
* ﻿[Virtualize Long List](https://legacy.reactjs.org/docs/optimizing-performance.html#virtualize-long-lists)﻿

**Hooks**

* ﻿[Built In Hooks](https://react.dev/reference/react/hooks) ﻿
  + useState
  + useEffect
  + useContext
  + useRef
  + useMemo
  + useCallback
  + useReducer
  + useActionState
  + useId
* ﻿[Custom Hooks](https://react.dev/learn/reusing-logic-with-custom-hooks#extracting-your-own-custom-hook-from-a-component)﻿
* ﻿[Pure Component And Side Effects](https://react.dev/learn/keeping-components-pure)﻿

**State Management and Routing**

* ﻿[State Management With useContext](https://react.dev/reference/react/useContext)﻿
* ﻿[When To Use Context](https://legacy.reactjs.org/docs/context.html)﻿
* ﻿[Redux](https://redux.js.org/introduction/getting-started)﻿
* ﻿[Redux Essentials](https://redux.js.org/tutorials/essentials/part-1-overview-concepts)﻿
* ﻿[Middleware](https://redux.js.org/understanding/history-and-design/middleware) ([Thunk](https://redux.js.org/usage/writing-logic-thunks" \t "_blank), Saga) **Good to Know**
* Redux toolkit[﻿](https://redux.js.org/usage/writing-logic-thunks)﻿
* ﻿[React Redux Hooks](https://react-redux.js.org/api/hooks#using-hooks-in-a-react-redux-app)﻿
* ﻿[Redux Devtool](https://chromewebstore.google.com/detail/redux-devtools-next/oamphgegmigmlgkdnijmeomjenbmkbdg)﻿

**Routing**

* ﻿[React Router](https://reactrouter.com/en/main/start/tutorial)﻿
  + Routes
  + Route
  + ﻿[Browser Router , Hash Router, Memory Router, Navigate Router](https://reactrouter.com/en/main/router-components/browser-router)﻿
  + URL Param
  + Nested Route
  + Lazy
  + Hooks
    - useNavigation
    - useHref
    - useNavigate
    - useParams
    - useOutlet
    - useRoutes
    - useSearchParams
    - useLocation
    - Share data to component on route change
      * Location, history
  + ﻿[Server Side Rendering](https://reactrouter.com/en/main/guides/ssr)

**Unit Testing and Type Checking**

* ﻿[Unit Testing: React Testing Library](https://testing-library.com/docs/react-testing-library/intro/)﻿
* ﻿[Jest: Mock Functions](https://testing-library.com/docs/dom-testing-library/api-events/#using-jest-function-mocks)﻿
* ﻿[Mock API, Mock fetch API](https://www.taniarascia.com/how-to-test-useeffect-api-call/)﻿

Type Checking

* What is [Prop-Types](https://www.npmjs.com/package/prop-types)﻿
* ﻿[React With TypeScript](https://react.dev/learn/typescript)

**Unit Testing with Jest & React Testing Library (RTL)**

* ﻿[What Is Unit Testing](https://www.geeksforgeeks.org/unit-testing-software-testing/)﻿
* ﻿[JS Unit Test](https://medium.com/@yunusemresalcan/unit-test-javascript-93ed9deb250c)﻿
* ﻿[How To Start Unit Testing](https://www.freecodecamp.org/news/how-to-start-unit-testing-javascript/)﻿
* ﻿[Why And When to Write Unit Tests](https://www.geeksforgeeks.org/unit-testing-software-testing/#what-are-the-benefits-of-unit-testing)﻿
* Unit Testing Frameworks
  + ﻿[Jest](https://jestjs.io/)﻿
  + ﻿[React Testing Library](https://testing-library.com/docs/react-testing-library/intro/)﻿
* ﻿[Stages Of Unit Tests](https://medium.com/the-ways-of-web/unit-test-the-3as-f4ccf2e5edb5)﻿
* Coverage Report [Code Coverage](https://dev.to/stevescruz/awesome-jest-tip-coverage-report-h5j)﻿
* What is Jest, How to use with JS and React
* What is RTL
* Render in Unit Test

**SDLC And Design Patterns**

* ﻿[Software Development Life Cycle (SDLC)](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/)﻿
* ﻿[Design Patterns](https://refactoring.guru/design-patterns)