

# Front-End Developer

# React.JS



Components can be easily reused



Easy to create UI test cases



Debugging is getting easy with specialized chrome extension



Fast when it comes to displaying a big amount of components

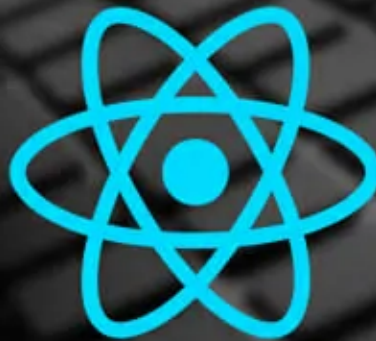


Seo friendly



React js virtual dom

JS



React

## Course Outline

# Web Fundamentals

- **Internet**

- How does the internet work
- What is HTTP
- Browsers and how they work
- DNS and how it works
- What is domain name
- What is hosting

- **HTML**

- Learn the basics
- Document structure
- Metadata
- Semantic HTML
- Heading and sections
- Text formatting basics
- Links
- List
- Navigation
- Tables
- Forms
- Validations in forms
- Image
- Audio
- Video
- Layouts

- **CSS**

- Box Model
- Selectors
- Cascade
- Color
- Layouts
- Flexbox
- Grid
- Spacing
- Borders
- Shadow
- Z-index Stacking contexts
- Functions





- Gradients
- Animation
- Filters
- Background
- Text & typography
  
- **Javascript**
  - Syntax and basic constructs
  - Variable declarations
    - Let, Var, Const
  - Variable name rule
  - Variable scope
    - Block, function, global
  - Data type
  - DOM manipulation
  - Learn AJAX calls
  - ES6+ and modular javascript
  - Hoisting
  - Event bubbling
  - Scope
  - Prototype
  - Shadow Dom
  - Strict
  - NPM
  - JSON
  - Loops and iterations
  - Control flow
  - Functions
  - Asynchronous javascript
    - Callback
    - Promise
    - Callback hell
    - Async await
    - SetTimeout
    - SetInterval

# ReactJS



- **Introduction to React**
  - To become aware of what React is
  - To be aware of developer tools available for React
  - To be able to set up the developer environment and a skeleton React application
  - To be aware of the security concerns with React
- **Components and JSX**
  - To understand what components are
  - To know what JSX is and why it is used in React JS
  - To be able to create Functional and Class Components
  - To be able to add multiple components
  - To know how to inspect components in the Browser
  - To understand how the in-browser tools work
- **Component Hierarchy**
  - To understand the development methodology suggested for React applications
  - To be able to identify the component hierarchy for a given application
- **Static Versions**
  - To be able to build a static version of an application
  - Understand how to use Props in components
  - To be able to test Component snapshots and Components with props
  - To be able to Mock components when testing
- **Identifying and Adding States**
  - How to identify state
  - Where state lives
  - Adding and Updating State to Components
  - How to add and update state to class and function components



