React JS

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js.

ReactJS History:

When compared to other technologies on the market, React is a new technology. Jordan Walke, a software engineer at Facebook, founded the library in 2011, giving it life. The likes of XHP, a straightforward HTML component framework for PHP, have an influence on React. React's newsfeed was its debut application in 2011. Later, Instagram picks it up and incorporates it into their platform.

Why we need React?

- Easy creation of dynamic applications: React makes it easier to create dynamic web applications because it requires less coding and offers more functionality, as opposed to JavaScript, where coding often gets complex very quickly.
- Improved performance: React uses Virtual DOM, thereby creating web applications faster. Virtual DOM compares the components' previous states and updates only the items in the Real DOM that were changed, instead of updating all of the components again, as conventional web applications do.
- Reusable components: Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their logic and controls, and they can be reused throughout the application, which in turn dramatically reduces the application's development time.
- Unidirectional data flow: React follows a unidirectional data flow. This means that when designing a React app, developers often nest child components within parent components. Since the data flows in a single direction, it becomes easier to debug errors and know where a problem occurs in an application at the moment in question.
- Small learning curve: React is easy to learn, as it mostly combines basic HTML and JavaScript concepts with some beneficial additions. Still, as is the case with other tools and frameworks, you have to spend some time to get a proper understanding of React's library.
- Easy Transition into mobile development: We already know that React is used for the development of web applications, but that's not all it can do. There is a framework called React Native, derived from React itself that is hugely popular and is used for creating beautiful mobile applications. So, in reality, React can be used for making both web and mobile applications.
- Dedicated tools for easy debugging: Facebook has released a Chrome extension that can be used to debug React applications. This makes the process of debugging React web applications faster and easier.

ReactJS Advantages

- React.js builds a customized virtual DOM. Because the JavaScript virtual DOM is quicker than the conventional DOM, this will enhance the performance of apps.
- ReactJS makes an amazing UI possible.
- Search - engine friendly ReactJS.
- Modules and valid data make larger apps easier to manage by increasing readability.
- Modules and valid data make larger apps easier to manage by increasing readability.

- React integrates various architectures.
- React makes the entire scripting environment process simpler.
- ReactJS is supported by a large community.

React Prerequisites

Here are some of the concepts that you should be familiar with, to one degree or another:

- Some familiarity with HTML & CSS.
- Basic knowledge of JavaScript & ES6.
- Node + NPM,
- GIT & CLI

How does React work?

React operates by creating an in-memory virtual DOM rather than directly manipulating the browser's DOM. It performs necessary manipulations within this virtual representation before applying changes to the actual browser DOM. React is efficient, altering only what requires modification.

Features of React

1. Component-Based Architecture

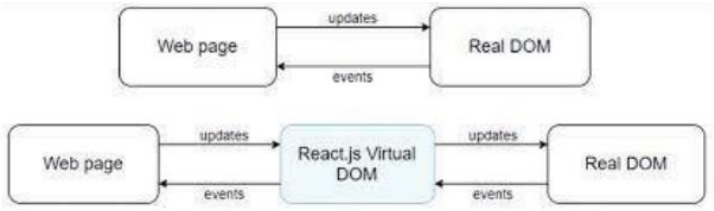
   React provides the feature to break down the UI into smaller, self-contained components. Each component can have its own state and props.

2. JSX (JavaScript Syntax Extension)

   JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript files. It makes React components more readable and expressive.

3. Virtual DOM

   React maintains a lightweight representation of the actual DOM in memory. When changes occur, React efficiently updates only the necessary parts of the DOM.



4. One-way Data Binding

   One-way data binding, the name itself says that it is a one-direction flow. The data in react flows only in one direction i.e. the data is transferred from top to bottom i.e. from parent components to child components. The properties (props) in the child component cannot return the data to its parent component but it can have communication with the parent components to modify the states according to the provided inputs.

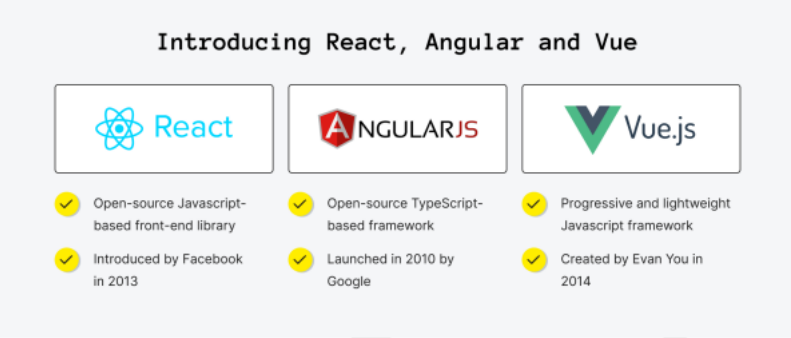
5. Performance

   As we discussed earlier, react uses virtual DOM and updates only the modified parts. So, this makes the DOM to run faster. DOM executes in memory so we can create separate components which makes the DOM run faster.

6. Components

React divides the web page into multiple components as it is component-based. Each component is a part of the UI design which has its own logic and design as shown in the below image. So the component logic which is written in JavaScript makes it easy and run faster and can be reusable.

Difference between React, Angular and Vue



Feature	Angular	React	Vue
Language	TypeScript	JavaScript	JavaScript
Rendering	Server-side, Client-side	Client-side	Client-side
Virtual DOM	Yes	Yes	Yes
Data binding	Two-way binding	One-way binding	Two-way binding
Templating	HTML-based	JSX-based	HTML-based
Learning curve	Steep	Moderate	Easy
Community	Large and active	Large and active	Rapidly growing
Performance	Slower than React, faster than Vue	Fast	Faster than Angular and React
Size	Large	Small	Small
Architecture	Full-fledged framework	UI library	Progressive framework
Use cases	Large-scale enterprise applications with complex requirements	Single-page applications, component-based UIs	Small to medium-sized applications, prototypes, and simple projects
Popular companies using it	Google, IBM, Microsoft, Deutsche Bank	Facebook, Instagram, Airbnb, Uber, Netflix	Alibaba, Xiaomi, Xiaomi Mall, Xiaomi Youpin, Bilibili