**ReactJS**

* React is a JavaScript Library.
* ReactJS is one of the most popular JavaScript front-end libraries which has a strong foundation and a large community.
* ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components.
* It is an open-source, component-based front end library which is responsible only for the view layer of the application.
* Today, most of the websites are built using MVC (model view controller) architecture. In MVC architecture, React is the 'V' which stands for view
* ReactJS is a library not a framework
* ReactJS focus on UI does not focus other aspects
* It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram. ReactJS is known for building single page applications.
* A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code.
* The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks.
* ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.
* ReactJS heavily relay on JavaScript
* ReactJS compose complex UIs from small and isolated pieces of code called “components”.
* Elements are the smallest building blocks of ReactJS apps.

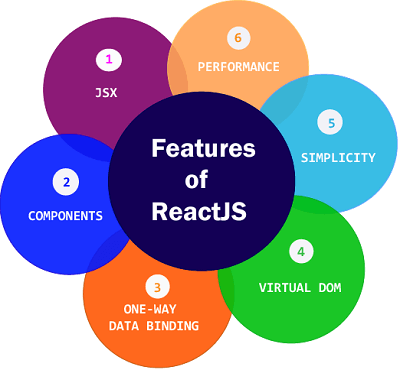
**Why we use ReactJS?**

* The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps.
* It uses virtual DOM (JavaScript object), which improves the performance of the app.
* The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks.
* It uses component and data patterns that improve readability and helps to maintain larger apps.

**Prerequisites**

Before learning ReactJS, you must have a good knowledge of JavaScript, HTML5, and CSS. The knowledge of ECMAScript 2015 or ES6 syntax can also be more helpful.

ReactJS Features



Currently, ReactJS gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as following.

* JSX
* Components
* One-way Data Binding
* Virtual DOM
* Simplicity
* Performance

**JSX**

JSX stands for JavaScript XML. It is a JavaScript syntax extension. It’s an XML or HTML like syntax used by ReactJS. This syntax is processed into JavaScript calls of React Framework. It extends the ES6 so that HTML like text can co-exist with JavaScript react code. It is not necessary to use JSX, but it is recommended to use in ReactJS.

**Components**

ReactJS is all about components. ReactJS application is made up of multiple components, and each component has its own logic and controls. These components can be reusable which help you to maintain the code when working on larger scale projects.

**One-way Data Binding**

ReactJS is designed in such a manner that follows unidirectional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application. If the data flow is in another direction, then it requires additional features. It is because components are supposed to be immutable and the data within them cannot be changed. Flux is a pattern that helps to keep your data unidirectional. This makes the application more flexible that leads to increase efficiency.

**Virtual DOM**

A virtual DOM object is a representation of the original DOM object. It works like a one-way data binding. Whenever any modifications happen in the web application, the entire UI is re-rendered in virtual DOM representation. Then it checks the difference between the previous DOM representation and new DOM. Once it has done, the real DOM will update only the things that have actually changed. This makes the application faster, and there is no wastage of memory.

**Simplicity**

ReactJS uses JSX file which makes the application simple and to code as well as understand. We know that ReactJS is a component-based approach which makes the code reusable as your need. This makes it simple to use and learn.

**Performance**

ReactJS is known to be a great performer. This feature makes it much better than other frameworks out there today. The reason behind this is that it manages a virtual DOM. The DOM is a cross-platform and programming API which deals with HTML, XML or XHTML. The DOM exists entirely in memory. Due to this, when we create a component, we did not write directly to the DOM. Instead, we are writing virtual components that will turn into the DOM leading to smoother and faster performance.

Pros and Cons of ReactJS

Today, ReactJS is the highly used open-source JavaScript Library. It helps in creating impressive web apps that require minimal effort and coding. The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. There are important pros and cons of ReactJS given as following:

Advantage of ReactJS

**1. Easy to Learn and USe**

ReactJS is much easier to learn and use. It comes with a good supply of documentation, tutorials, and training resources. Any developer who comes from a JavaScript background can easily understand and start creating web apps using React in a few days. It is the V(view part) in the MVC (Model-View-Controller) model, and referred to as ?one of the JavaScript frameworks.? It is not fully featured but has the advantage of open-source JavaScript User Interface(UI) library, which helps to execute the task in a better manner.

**2. Creating Dynamic Web Applications Becomes Easier**

To create a dynamic web application specifically with HTML strings was tricky because it requires a complex coding, but React JS solved that issue and makes it easier. It provides less coding and gives more functionality. It makes use of the JSX(JavaScript Extension), which is a particular syntax letting HTML quotes and HTML tag syntax to render particular subcomponents. It also supports the building of machine-readable codes.

**3. Reusable Components**

A ReactJS web application is made up of multiple components, and each component has its own logic and controls. These components are responsible for outputting a small, reusable piece of HTML code which can be reused wherever you need them. The reusable code helps to make your apps easier to develop and maintain. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

**4. Performance Enhancement**

ReactJS improves performance due to virtual DOM. The DOM is a cross-platform and programming API which deals with HTML, XML or XHTML. Most of the developers faced the problem when the DOM was updated, which slowed down the performance of the application. ReactJS solved this problem by introducing virtual DOM. The React Virtual DOM exists entirely in memory and is a representation of the web browser's DOM. Due to this, when we write a React component, we did not write directly to the DOM. Instead, we are writing virtual components that react will turn into the DOM, leading to smoother and faster performance.

**5. The Support of Handy Tools**

React JS has also gained popularity due to the presence of a handy set of tools. These tools make the task of the developers understandable and easier. The React Developer Tools have been designed as Chrome and Firefox dev extension and allow you to inspect the React component hierarchies in the virtual DOM. It also allows you to select particular components and examine and edit their current props and state.

**6. Known to be SEO Friendly**

Traditional JavaScript frameworks have an issue in dealing with SEO. The search engines generally having trouble in reading JavaScript-heavy applications. Many web developers have often complained about this problem. ReactJS overcomes this problem that helps developers to be easily navigated on various search engines. It is because React.js applications can run on the server, and the virtual DOM will be rendering and returning to the browser as a regular web page.

**7. The Benefit of Having JavaScript Library**

Today, ReactJS is choosing by most of the web developers. It is because it is offering a very rich JavaScript library. The JavaScript library provides more flexibility to the web developers to choose the way they want.

**8. Scope for Testing the Codes**

ReactJS applications are extremely easy to test. It offers a scope where the developer can test and debug their codes with the help of native tools.

Disadvantage of ReactJS

**1. The high pace of development**

The high pace of development has an advantage and disadvantage both. In case of disadvantage, since the environment continually changes so fast, some of the developers not feeling comfortable to relearn the new ways of doing things regularly. It may be hard for them to adopt all these changes with all the continuous updates. They need to be always updated with their skills and learn new ways of doing things.

**2. Poor Documentation**

It is another cons which are common for constantly updating technologies. React technologies updating and accelerating so fast that there is no time to make proper documentation. To overcome this, developers write instructions on their own with the evolving of new releases and tools in their current projects.

**3. View Part**

ReactJS Covers only the UI Layers of the app and nothing else. So you still need to choose some other technologies to get a complete tooling set for development in the project.

**4. JSX as a barrier**

ReactJS uses JSX. It's a syntax extension that allows HTML with JavaScript mixed together. This approach has its own benefits, but some members of the development community consider JSX as a barrier, especially for new developers. Developers complain about its complexity in the learning curve.

**Components**

* Components are like JavaScript functions
* Component is a simple JavaScript file
* Components are re-usable
* Components are invoked one another
* Each component specifies as a custom html tag
* A component describes what you want to see on the screen
* Components accept inputs (called “props”) and return ReactJS elements(UI) describing what should appear on the screen.
* Components are two Types
* Functional Components
  + are just JavaScript functions
  + accept input as props and return html as UI
* Class Components
  + must contain render method
  + class components are basically es6 classes
  + accept input as props and return html as UI
  + maintains state (private to that component)
  + provide life cycle hooks
  + are statefull/smart/container

Difference between Function and Class components

|  |  |
| --- | --- |
| Function | Class |
| Simple Functions | More feature rich |
| Use function components as much as Possible | Maintain their own private data-state |
| Absence of this keyword | Complex UI Logic |
| Solution without using state | Provide life cycle hooks |
| Mainly responsible for UI |  |
| stateless/dumb/presentational | stateful/smart/container |
|  |  |

**JSX (Extension to JavaScript)**

* JSX stands for extension to JavaScript
* JSX make ReactJS code simpler and elegant
* JSX transpire into JavaScript
* JSX Difference
  + class-className
  + for-htmlFor
  + camel case naming conventions
    - onclick-onClick
    - tabindex-tabIndex

**Props**

* Allow the component dynamic
* props is an object that contain attributes and values
* props make component as re-usable
* props are immutable (i.e. value cannot be changed)
* to use props in class component use {this.props.prop-name}

**State**

* It is an object and privately maintained by component
* state can be changed with in the component
* we never modify the state directly (use setstate () method to modify state value)
* we can bind state values with in the render() function.
* UI should re-render when state has to change.
* ReactJS is not re-render the component when state values have changed directly.
* State values should not have modified directly, use setState() method to modify the state values.
* Call to setState() are asynchronous
* setState()
  + Always make use of setState and never modify the values directly
  + Code has to be executed after state has been updated? Place that code in the call back function which is the second argument to the setState method
  + When you have to update state based on the previous state value, pass in a function as an argument instead of the regular object.

|  |  |
| --- | --- |
| Props | State |
| Props get passed to the component | State is managed with in the component |
| Passed as function parameters | Variables declared in the function body |
| Props are immutable | State |
| Usage:  props-Functional Components  this.props-Class Components | Use stateHook-Functional Components  This.state-Class Components |

**List-Keys**

* A key is a special string attribute you need to include when you create list of elements
* keys give the elements a stable identity
* keys help ReactJS identify which items have changed, are added, or are removed
* helps in efficient update of the user interface
* Index as Key (When to use index as key)
  + The item in your list do not have unique id.
  + The list is a static list and will not change.
  + The list never be reordered and filtered.

**Styling ReactJS Components**

* Css Stylesheets
* inline styling
* CSS Modules
* CSS in JS Libraries

**Form Controls**

* change handler is required to update the state props
* event. preventDefault ()-prevent default behavior of form submission

Extra Note:

* To render a ReactJS element into a root DOM node, pass both to ReactJSDOM.render()
* In JavaScript classes, you need to always call super when defining the constructor of a subclass. All ReactJS component classes that have a constructor should start with a super(props) call.
* In ReactJS a single page divide into many components.
* In ReactJS we have default exports and named exports.
* ReactJS can return only one element
* if-else statements with in json is not valid.
* rce-snippet for class component
* rconst-snippet for constructor