

What is ReactJS?

ReactJS is a frontend Javascript library for Web Development.

Rendering elements in the dom

Elements are the smallest building blocks of React apps.

It's like a small entity of the components. Components are made of elements.

There is a single element with the id = "root" in index.html file through which all the components and the elements in components get injected to that root div.

It is basically like a entry point of the react application

We have to pass root element to which is actually dom to ReactDOM.createRoot() and then pass react element to root.render();

How does react render and update the UI?

React DOM compares the element and its children to the previous one and only applies DOM updates which are necessary to bring dom to desired changes.

Even if we recreate element forcefully for demonstration then, only the text which is changed gets updates on the real dom

What are React Components?

React components are the reusable block of codes just like javascript functions. The only difference is that they return HTML code. There are two types of components which are class components and functional components.

Functional Components vs Class Components?

Functional Components:

Functional Components are the same as javascript functions. These are the most common components across react.

Class Components:

Most of the modern web apps are based on these components. These are like simple classes containing multiple functions. These are complicated as compared to the functional components.

Functional Components	Class Components
A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).	A class component requires you to extend from React. Component and create a render function which returns a React element.
Hooks can be easily used in functional components to make them Stateful.	It requires different syntax inside a class component to implement hooks.

example: <code>const [name,SetName]= React.useState('')</code>	example: <code>constructor(props) { super(props); this.state = {name: ''} }</code>
--	--

What are Props?

Props are arguments are passed to the react components

While calling the component the props are passed to the components using HTML attributes.
`<Student name="shridhar"/>`

What is the state in React?

State is a built-in object in react which is used to store the data or information of a particular component. Whenever the state of a component changes, the component re-renders.

Props vs State in React.

Props	State
Props (properties) are configuration of the components. They are received from above and are immutable	The state is a data structure that starts with the default value when a component mounts

Type checking prop types:

React has some built in type checking functionalities. For that we have to import PropTypes like array, bool, number, string, object etc.

Handling Events in React:

It is very similar to handling events in dom. We can pass the function name as the event handler to the event attribute

`<button onClick = {activateUser}> Activate User </button>`

React Lifecycle methods of class components:

Mounting - putting element to dom

Updating - component is updated whenever the there is change in components state or props

Unmounting - to remove elements from dom