**1. React Render HTML**

* In React, we "render" HTML using JavaScript code that translates the structure of our interface into the HTML elements shown in the browser.
* Rendering is done through the ReactDOM.render() method, which takes in a React component and attaches it to a DOM node.
* This is how React inserts the application’s HTML structure into the DOM, updating it efficiently when data changes.

**2. React JSX**

* JSX (JavaScript XML) is a syntax extension that allows writing HTML elements directly within JavaScript code.
* JSX makes it easier to visualize UI components by embedding HTML-like syntax. For example, <h1>Hello, World!</h1> can be written directly in JavaScript.
* Behind the scenes, JSX compiles to React.createElement() calls, which create virtual DOM nodes, making it a key feature for building React components.

**3. React Components**

* Components are the building blocks of a React application, enabling the UI to be divided into independent, reusable pieces.
* Components can be **functional** (written as JavaScript functions) or **class-based** (using ES6 classes).
* Each component can maintain its own state and lifecycle, making it possible to build complex UIs by combining smaller, self-contained components.

**4. React Class**

* A **class component** in React is a JavaScript class that extends React.Component.
* Class components can hold and manage their own state, unlike functional components (before React introduced hooks).
* They also have access to lifecycle methods (e.g., componentDidMount, shouldComponentUpdate), which control what happens during the component’s lifecycle.

class MyComponent extends React.Component {

render() {

return <h1>Hello, Class Component!</h1>;

}

}

Practice example’s

import React from 'react';

import ReactDOM from 'react-dom';

const element = <h1>Hello, World!</h1>;

ReactDOM.render(element, document.getElementById('root'));

import React from 'react';

import ReactDOM from 'react-dom';

function Greeting() {

return <h2>Welcome to React!</h2>;

}

ReactDOM.render(<Greeting />, document.getElementById('root'));

import React from 'react';

import ReactDOM from 'react-dom';

function Header() {

return <h1>This is the Header</h1>;

}

function Footer() {

return <h3>This is the Footer</h3>;

}

ReactDOM.render(

<div>

<Header />

<Footer />

</div>,

document.getElementById('root')

);

JSX

import React from 'react';

const element = (

<div>

<h1>Hello, React JSX!</h1>

<p>This is an example of JSX syntax.</p>

</div>

);

import React from 'react';

const name = "Ankit";

const greeting = <h1>Hello, {name}!</h1>; // Embeds JavaScript variable into JSX

import React from 'react';

const style = {

color: 'blue',

fontSize: '20px'

};

const element = <h1 style={style}>Styled Text</h1>;

components

import React from 'react';

function Welcome() {

return <h1>Welcome to React Components!</h1>;

}

import React from 'react';

function Greeting(props) {

return <h1>Hello, {props.name}!</h1>;

}

import React from 'react';

function Profile() {

return <h1>User Profile</h1>;

}

function UserDetails() {

return (

<div>

<Profile />

<p>This is user information.</p>

</div>

);

}

Class

import React, { Component } from 'react';

class HelloWorld extends Component {

render() {

return <h1>Hello from Class Component!</h1>;

}

}

import React, { Component } from 'react';

class Greeting extends Component {

render() {

return <h1>Hello, {this.props.name}!</h1>;

}

}

import React, { Component } from 'react';

class Counter extends Component {

constructor() {

super();

this.state = { count: 0 };

}

increment = () => {

this.setState({ count: this.state.count + 1 });

};

render() {

return (

<div>

<p>Count: {this.state.count}</p>

<button onClick={this.increment}>Increment</button>

</div>

);

}

}