REACT-JS SYNTAXS

Creating Elements using React JS:

REACT CDN: <script src="https://unpkg.com/react@17.0.0/umd/react.development.js"></script>

<script src="https://unpkg.com/@babel/standalone@7.12.4/babel.js"></script>

THE REACT TYPE ATTRIBUTE VALUE OF THE HTML SCTIPT ELEMENT SHOULD BE MODULE TO RUN REACT.JS.

React.createElement() method is used to create an element using React-js.It's similar to the document.createElement() method in regular Javasctipt.

React.createElement(type, props);

TYPE – TAG NAME LIKE div, h1, p, etc.

PROPS – PROPERTIES LIKE className, onClick, id, etc.

```
<script type="module">

    const elementProps = {className: "greeting", children: "Hello world!"};

    const elementType = "h1";

    const element = React.createElement(elementType, elementProps);

</script>
```

ReactDOM.render(): The ReactDOM.render() method is used to display the React element.

ReactDOM.render(reactElement, container);

ReactElement – What to Render

Container – Where to render

JSX (JAVASCRIPT XML): React JS introduced a new HTML like syntax named JSX to create elements.

CONST element = <h1 className=”greeting”>Hello World</h1>;

THE ABOVE JSX ELEMENT COMPILES TO,

CONST elementProps = { className: “greeting “, children: “ Hello World ” };

CONST element = React.createElement(“h1”, elementProps);

DIFFERENT BETWEEN HTML AND JSX:

HTML	JSX
class	className
for	htmlFor

COMPONENT: A Component is a JS function that returns a JSX element.

```
const Welcome = () => <h1 className="message">Hello, User</h1>;
```

PROPERTIES(PROPS): React allows us to pass information to a component using props.

```
<Component propName1="propValue1" propName2="propValue2" />
```

WEBPACK: Stitches together a group of modules into a single file (or group of files). This process is called Bundling.

KEYS: help React to identify which items have changed, added, or removed. They should be given to the elements inside the array for a stable identity.

```
const userDetails = [{

   uniqueNo: 1,

   imageUrl: 'https://assets.ccbp.in/frontend/react-js/esther-howard-img.png',

   name: 'Esther Howard',

   role: 'Software Developer'

 }

]
```

COMPONENTS: THERE ARE TWO WAYS TO WRITE REACT COMPOONENTS.

FUNCTIONAL COMPONENTS: These are JavaScript functions that take props as a parameter if necessary and return react element (JSX).

```
const Welcome = () => <h1>Hello, User</h1>;

export default Welcome;
```

CLASS COMPONENTS: These components are built using an ES6 class. To define a React Class Component,

CREATE AN ES6 CLASS THAT EXTENDS React.Component.

ADD A SINGLE EMPTY METHOD TO IT CALLED render ().

EXTENDS: THE extends KEYWARD IS USED TO INHERIT METHODS AND PROPERTIES FROM THE React.Components.

render (): method is the only required method in a class component. It returns the JSX element.

```
import { Component } from "react";

class MyComponent extends Component {

  render() {

        return JSX;

  }

}
```

DEFAULT PROPS: is a property in React Component used to set default values for the props. This is similar to adding default parameters to the function.

```
ComponentName.defaultProps = {

  propName1: "propValue1",

  propName2: "propValue2"

}
```

setState() Object Syntax: object syntax can be used while updating the state to the value that is independent of the previous state.

```
this.setState(

  {propertyName1: propertyValue1},

  {propertyName2: propertyValue2}

            // and many more...

);
```

INPUT ELEMNTS: In React, the Input Element value can be handled in two ways:

CONTROLLED INPUT: If the Input Element Value is handled by a React State then it is called CONTROLLED INPUT. Controlled Inputs are the React Suggested way to handle Input Element value.

UNCONTROLLED INPUT: If the Input Element Value is handled by the browser itself.

Example: `<input type="text" />`

DEBUGGING: Debugging is the process of finding & fixing the bugs in the code. We can debug using:

BROWSER DEVELOPER TOOLS: These are the tools provided by the browsers to debug the application loaded in the web browser. Using Browser Developer Tools, we can:

VIEW THE SOURCE CODE (HTML, CSS, JS),

VIEW AND CHANGE CSS, RUN JAVASCRIPT IN THE CONSOLE,

VIEW LOGGED MESSAGES IN THE CONSOLE,

CHECK THE RESPONSIVENESS OF AN APPLICATION.

REACT DEVELOPER TOOLS: For React Developer Tools, we can install the REACT DEVELOPER TOOLS EXTENSION for Google Chrome.

COMPONENT LIFE CYCLE PHASES: Every React Component goes through three phases throughout its lifetime:

MOUNTING PHASE: In this phase, the instance of a component is CREATED and INSERTED into the DOM.

(1) CONSTRUCTOR (): This method is used to set up the INITIAL STATE and CLASS VARIABLES.

```
constructor(props) {

  super(props)

  this.state = { date: props.date }

}
```

(2) RENDER (): method is used to return the JSX that is displayed in the UI.

(3) COMPONENTDIDMOUNT (): method is used to run statements that require that the component is already placed in the DOM.

Example: set timers, initiate API calls, etc.

UPDATING PHASE: In this phase, the component is updated whenever there is a change in the component's state.

RENDER (): method is called whenever there is a change in the component's state.