# ReactJS Interview Prep:

**What are the features of REACT?**

* **JSX**: JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.
* **Components**: Components are the building blocks of any React application, and a single app usually consists of multiple components. It splits the user interface into independent, reusable parts that can be processed separately.
* **Virtual DOM**: React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, virtual DOM changes only that object in the real DOM, rather than updating all the objects.
* **High performance**: React updates only those components that have changed, rather than updating all the components at once. This results in much faster web applications.

**What is JSX?**

* JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.

**Can Web browsers read JSX directly?**

* Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects and JSX is not a regular JS object.
* For a web browser to read a JSX file, the file needs to be transformed into a regular JS object. For this, we use Babel.

**What is the virtual DOM?**

* DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the tree ends in a node, and each node contains objects.
* React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects.

**Why use React instead of other frameworks like Angular?**

* Easy creation of dynamic applications: React makes it easier to create dynamic web applications because it provides less coding and provides more functionality, whereas, with JavaScript applications, code tends to get complex very quickly.
* Improved Performance: React uses virtual DOM, which makes web applications perform faster. Virtual DOM compares its previous state and updates only those components in the real DOM, whose states have changed, rather than updating all the components – like conventional web applications.
* Reusable components: Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their own logic and controls, and they can be reused through the application, which, in turn, dramatically reduces the development time of an application.
* Dedicated tools for easy debugging: Facebook has released a chrome extension that we can use to debug React applications. This makes the process of debugging React to web applications faster and easier.

**What is the difference between ES6 and ES5 standards?**

=> These are the few instances where ES6 syntax has changed from ES5 syntax.

* Require to import
* Exports vs export

**How do you create a react app?**

=> These are the steps for creating a React app:

* Install NodeJS on the computer because we need npm to install the React library. NPM is the node package manage that contains many JS libraries, including React.
* Install the create-react-app package using the command prompt or terminal.
* Install a text editor of your choice, like VS code or any other

**What is an event in React?**

* An event is an action that a user or system may trigger, such as pressing a key, click of a mouse, etc.
* React events are named using camelCase, rather that lowercase in HTML.
* With JSX, you pass a function as the event handler, rather than a string in HTML.

**How do you create an event in React?**

* A React event can be created by doing the following

const function = () => {

const clickHandler = () =>{

console.log(“That worked!!!”)

}

return (

<button onClick={clickHandler}>Click Me!</button>

)

}

**What are synthetic events in React?**

* Synthetic events combine the response of different browser’s native events into one API, ensuring that the events are consistent across different browsers.
* The application is consistent regardless of the browser it is running in. Here, preventDefault is a synthetic event.

**Explain how lists work in React?**

* We create lists in React as we do in regular JavaScript. Lists display data in an ordered format
* The traversal of lists is done using the map() function.

Ex:

const names = [‘Kohli’, ‘Sachin’, ‘Dhoni’, ‘Jadeja’]

const listOfNames = () => {

const listItems = names.map((name) =>

<li key={name}>

{name}

</li>

);

return (

<ul>{listItems}</ul>

)

}

**Why is there a need for using keys in Lists?**

* Keys are very important in lists for the following reasons:
* A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists.
* It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered.

**What are forms in React?**

* React employs forms to enable users to interact with web applications.
* Using forms, users can interact with the application and enter the required information whenever needed. Form contains certain elements, such as text fields, buttons, check-boxes, radio buttons, etc.
* Forms are used for many different tasks such as user authentication, searching, filtering, indexing, etc.

**How do you create forms in React?**

* We create forms in React by doing the following

const NameForm = () => {

const [name, setName] = useState(‘’)

const changeHandler = (e) => {

setName(e.target.value)

}

const submitHandler = (event) => {

event.preventDefault();

alert(‘A name was entered’ + name)

}

return (

<form onSubmit={submitHandler}>

<label>

Name:

<input type=’text’ value={name} onChange={onChangeHandler} />

</label>

</input>

<button type=’submit’>Submit</button>

</form>

)

}

* The above code will yield an input with the label Name and a submit button. It will also alert user when the submit button is pressed.

**What are the components in React?**

* Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately.

**What is the use of render() in React?**

* It is required for each class-based component to have a render() function. This function returns the HTML, which is to be displayed in the component.
* If you need to render more than one element all of the elements must be inside one parent tag like <div>, <form>.

**What is state in React?**

* The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.
* The change in state can happen as a response to user action or system-generated events. It determines the behavior of the component and how it will render.