Pre-requiste for React

HTML:elements,tags,attributes

css:general Properties,selectors,flext,grid

JS: Varibales,Data types,Function,Operators,selection statements:if else switch

object,array,DOM and Dom manipulation,Events

Advance Js:ES6,Promises,oops,let var const,Module scope,clousers

What is React or React JS ?

-React JS is a JavaScript Library

-React Js is not a framework

What is Library?

-Library is a collection of pre-defined functions,classes or Objects

-React is a collection of pre-defined function,classes and Objects

What is the purpose of ReactJS?

-React JS is used to create UserInterface

-If we have to create frontend application then it wont possible by react we need other libraries like ReactDOM,Redux,Axios,ReactRouter

React==>UserInterface==>+==>Other Libraries==>FrontEnd application(Frontend Engineers)

What is UserInterface?

-Collection of UI Components

We can create UI with HTML,JS but with React JS?

-rendering,siglepage,etc...

Create a simple Button Using HTML

Button Element-----Button Tag



Create a simple button using JS

Pre-defined object and functions

DOM Button Element

DOM

Creating UI(Button) is easy on HTML In Html we cannot reuse the UI

Creating UI in JS looks little complex In js we can reuse the UI

So both html and js have adv and dis to overcome both we use reactjs

React

In React, creating UI is easy and simple like html

In react, we can reuse as UI multiple times

Why react is used to create user interface?

1.Creating UI is simple in react

2. We can re-use the UI in the react

Creating simple using React

1.Create a html file

2. Add basic code of html

3.add script tag inside body so that we can write react code

4.Integrate reactjs with html document

HTML-------------HTML elements-------------------UI Comp

JS---------Predefined object and function--------DOM element------add in the dom-------UI

React-------Predefined object and functions----------react element add in the dom-------ui

5.Create React Element

Var reactele=.createElement(<tagName>,{--},children)

Children means html children is body and body children is tittle etc

6.Add react element inside the DOM

ReactDOM is a library which is used to add React Elements inside the DOM

Integrate ReactDOM with HTML Document

ReactDOM.render(<React element>,<DomElementReference>);

Creating UI in reactjs is simple or complex?

-Creating UI components is complex because of createElement same as js so JSX (same as HTML) came into picture

ReactDev----JSX Element-----babel compilers----ReactElements---ReactDOM---DOM---UI

HTML JSX

<h2>----</h2> var h2=<h2>---</h2>

<button>--</button> var button=<button>----</button>

How to configure attributes in react?

A)Second argument in React.createElement({----}); is used for configuring attributes

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Task1</title>

    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

</head>

<body>

    <div id="root"></div>

    <script>

        let div=React.createElement('div',null,

        React.createElement('div',null,

        React.createElement('div',null,

            [React.createElement('h2',{},"React Elements"),

             React.createElement('p',{},"Reacts look very complex now")

            ]

        )

        ));

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

    </script>

</body>

</html>

Complex div inside div inside div inside p and h2 tags

Dev---React---ReactElements—DOM using ReactDOM---UI

Dev---JSX Elements----babel Compiler(convert jsx to react elements) ----ReactElements---ReactDOM---DOM—UI

-Use of Babel compiler is converting jsx to react elements

Note: We need to add script tag of babel and need to add  <script type="text/babel">.

How to add Javascript code inside the jsx element?

Syntax: <jsx-element>{javascript code}</jsx-element>

{} is mandatory

Eg: let container=(

            <div>

            {card}

            {card}

            </div>

        );

-Create functions for passing arguments and easy reusing of component

In react, whenever we want to create any UI then, that UI we will always create in a javascript function.

Why we have to create UI inside the javascript function?

1. We can re-use the UI
2. We can pass dynamic data to the UI

Function createCard(){

Retrun <div></div>

}

createCard();

Instead of calling functions like this In react js we can call like element tag

Eg: <CreateCard></CreateCard> or <CreateCard/>

But the first letter should be captical.

Normal calling:

function multipleContainer(){

            return (

                <div>

                    {createContainer("React JS")}

                    {createContainer("Angular")}

                    {createContainer("Node JS")}

                </div>

            );

        }

Element tag based function:

 function MultipleContainer(){   //first letter should be capital

            return (

                <div>

                    <CreateContainer/>

                    <CreateContainer/>

                    <CreateContainer/>

                </div>

            );

        }

Functional Component:

-It is a simple javascript function whose name starts with uppercase and which returns UserInterface(JSX or React Elements).

There are two types of Component in React

1. Functional Component (Mostly used)
2. Class Component (Not recommended to use)

Components represents view on the screen or on the webpage

React application is a collection of components.

Check TechContainer.html for example

Note:

Any component which we assign to ReactDOM, that component is considered as a parent component for entire application.

Eg: ReactDOM.render(<App/>,document.getElementById('root'));

Rendering = calling

Rendering=calls+converted to react elements +added in the dom

What is the difference between normal js and react functional component?

1. RFC should always starts its name with uppercase
2. Normal js function can start names with any case
3. RFC will always return jsx code
4. NJF can return any type of value
5. RFC contains JSX code
6. NJF does not contain any jsx code

Note:

Using Attribute Concept we can pass arguments to the components if it is rendered in the element or Tag format

Props:

Props are the properties of component

Props are used to pass dynamic data or input data to the components

Props are used to share the data from parent comp to child compoent

Eg:

 function CreateCard(props){  //data={title:" ",image:" "} props stored

            //JSX code to create a card

            return (<div class="card">

                    <img src={props.image} width="100%" height="210px"/>

                    <h3>{props.title}</h3>

                    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aspernatur dolore, obcaecati totam veritatis quo maiores. Voluptate quidem deleniti rem iste sint sunt facilis culpa ab, iusto nostrum iure provident. Atque.</p>

                    <button>Profile Details </button>

                </div>);

        }

Always first parameter is object it will store all the parameters passed through the function

eg:

<CreateCard title="Nikhil Thula" image="https://media.istockphoto.com/id/1391718981/photo/portrait-of-a-confident-young-businessman-standing-with-his-arms-crossed-in-an-office.jpg?s=612x612&w=0&k=20&c=eF\_0QCtw-Y8Q2c4\_xQe6KTkcSPiGCT6qBf6nuavE2Dg="/>

We need to pass function CreateCard(props){  //data={title:" ",image:" "} props stored like this

<img src={props.image} width="100%" height="210px"/>

Note:

Functional component first parameter will be always object, this object stores props or this object contains set of props as a properties.

Tools used to develop ReactApplication

1.CRA – it takes to time to create application slow

2. Vite🡪Basic React Application

Installed

Pre-Configured

Easy and simple to install configure other libraries

Optimised code for production

Development server

Start,stopand loading appp or project inside this dev server

url—to access our application in any browser

Create a basic React application using vite