REACT JS TRAINING

DAY 1 : Introduction to React JS

Q1. What is React JS?

React JS is a popular JavaScript library used for building user interfaces for web applications developed by FB developers. React JS is not a framework.

Q2. Perquisites to learn and implement React JS?

* Should have knowledge about JS.
* Know about ES6 is an advantage.
* Use Visual Studio Code to create programs.

Q3. Difference between Library and framework?

**Library** is a collection of pre-written code that provides specific functionality or features. It typically consists of a set of functions, classes, or modules that developers can use to perform specific tasks. When using a library, the control flow of the application remains in the hands of the developer, who decides when and how to use the library's functionalities.

**Framework** is a software platform which have structured architecture and a set of rules, conventions, and tools for building application. When using a framework, the control flow of the application not in the hands of the developer. Developers work within the framework's predefined structure and utilize its provided tools and features.

=> Now, before start learning React JS, we first create a simple program with java script

<html>

<head>

<title>Page Title</title>

</head>

<body>

<div id="root"></div>

<script>

const target = document.getElementById("root");

const heading = document.createElement("h1");

heading.textContent = "This is test file.";

target.appendChild(heading);

</script>

</body>

</html>

Q4. What are the ways to include React in our application?

There are two ways to include React in our application :-

1. With the help of CDN (Content Delivery Network).
2. With the help of import packages.

Now, firstly in the above program we include react with the help of CDN

<html>

<head>

<title>Hello react</title>

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

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

</head>

<body>

<div id="root"></div>

<script>

const target = document.getElementById("root");

const heading = document.createElement("h1");

heading.textContent = "This is the dynamically added content.";

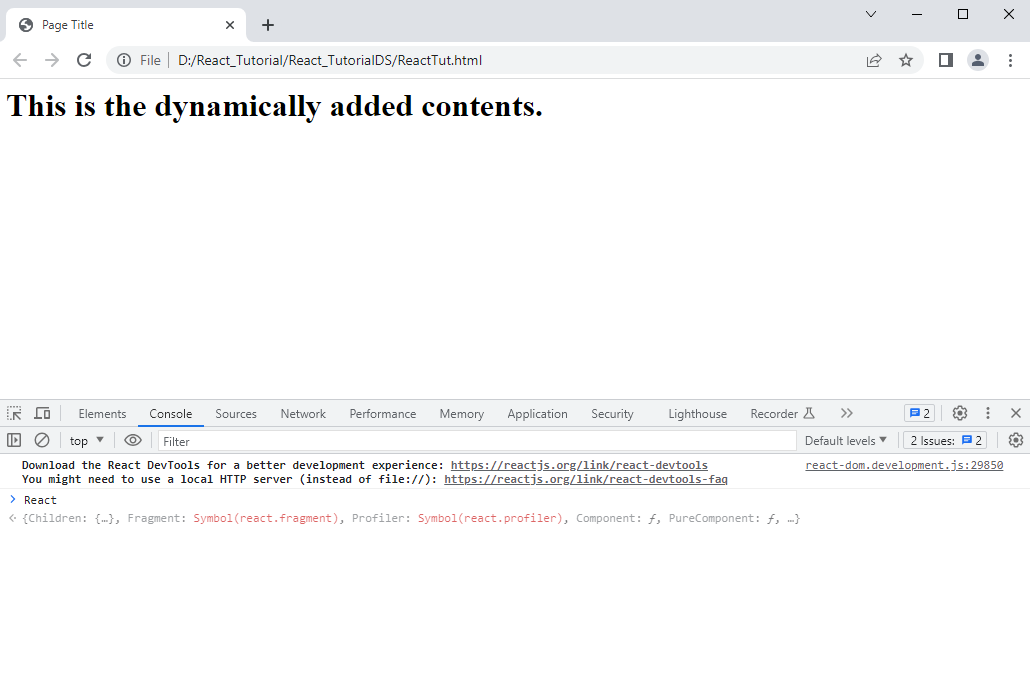
target.appendChild(heading);

</script>

</body>

</html>

In the above program we have added CDN link in head tag, this will help to include react in our program.



Q.5 Now the question arises that what is the smallest program of react js?

Above program is the smallest program of react JS because without writing a single line code of react JS, we can access the react in console. As you can see in above image.

Q.6 What is cross-origin origin?

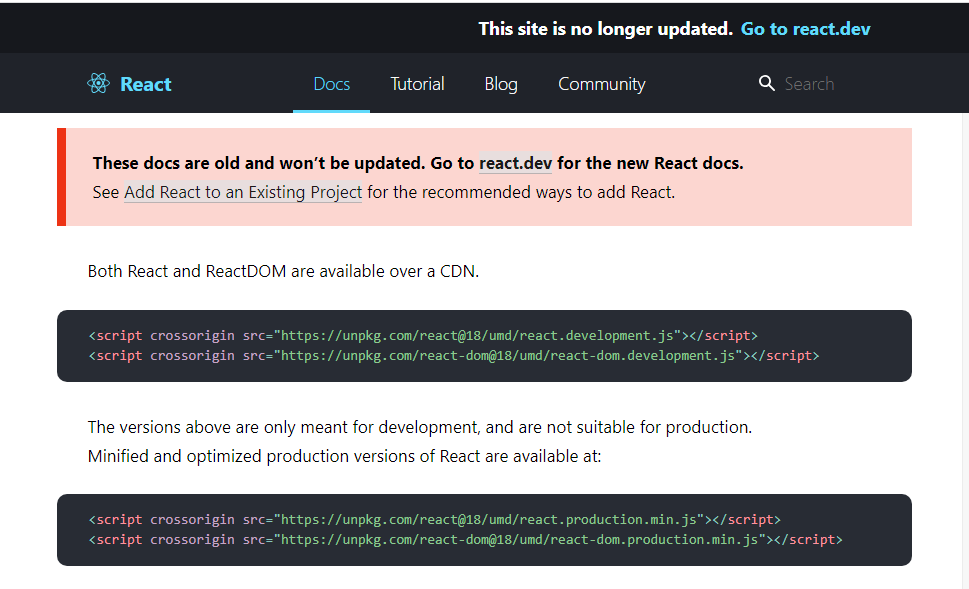
Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.

Basic React JS command and Folder structure

https://www.c-sharpcorner.com/blogs/reactjs-commands-and-folder-structure

=> URL to get CDN links to include react js in our application

<https://legacy.reactjs.org/docs/cdn-links.html>



In the above image you can see there are 2 types of links are present: -

1. Above links are for development, which means this include the expand version of react js which is in development and can further modified.
2. Below links are for production, which means this include the minified version of react js which are not in development, so these are best to use for projects.

------------------------------------------------------End of Day 1st training-----------------------------------------------------

DAY 2 : JavaScript code with the help of React js

<html>

<head>

<title>Page Title</title>

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

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

</head>

<body>

<div id="root"></div>

<script>

const heading1 = React.createElement(

'h1',

{ id: 'headingFirst', className: 'headingFirstClass' },

'Hello, This is first heading');

const heading2 = React.createElement(

'h1',

{ id: 'headingSecond', className: 'headingSecondClass' },

'Hello, This is Second heading');

const mydv = React.createElement(

'div',

{ id: 'headingThird', className: 'headingThirdClass' },

[heading1,heading2]);

const root=ReactDOM.createRoot(document.getElementById("root"));

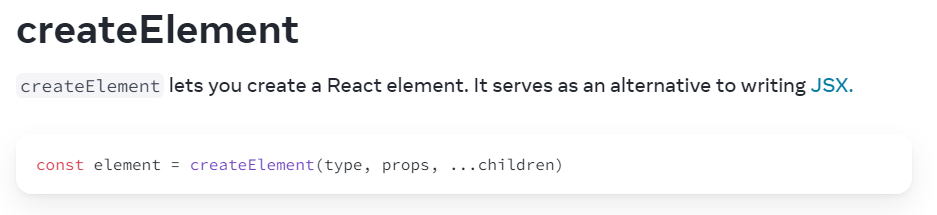
root.render(mydv);

</script>

</body>

</html>

=> In the above program, we have use React.createElement()



In above image createElement have three parameters: -

1. Type: we pass html elements like h1, h2, div, etc.
2. Props: we pass html attributes like id, classname, style, etc.
3. Children: we pass inner html or content.

------------------------------------------------------End of 2nd Day training---------------------------------------------------

DAY 3 : Install packages to include react in our project

1. Install Node.js: before run commands we have to install node.js in our system
2. Command: - npm init-y

Use: This command uses to initialize package manager and -y is used for skip questions.

1. Command: - npm install -d parcel

Use: this command is used to install parcel n -d is use for development.

1. npm install react

Use: this command is used to install react.

1. npm install reactdom

Use: this command is used to install react dom.

Above mentioned all commands are in proper sequence, so these commands should run in above sequence.

Q.1 What is npm?

npm is a package manager which is used to install packages. npm is the world's largest software registry. Open-source developers from every continent use npm to share and borrow packages, and many organizations use Command to run on CMD.

Q.2 What is a Parcel?

Parcel is a bundler. Performs many background operations like cache management, js minification, creating servers, providing hosts. etc.

Q.3 What is bundler?

Bundler is tool which do background operations like optimization, minification, providing host, etc.

Q.4 Can we deploy our website to server without bundler?

Yes, but all the background operations should be done by the user, these background operations are very time consuming so we use bundler for this purpose.

Q.5 What are package?

A package is a directory, package refers to a collection of related files, modules, or resources that are bundled together for organizational or distribution purposes. Packages are used to group and encapsulate code, allowing developers to easily manage dependencies, share code, and organize their projects.

Q.6 What are modules?

In JavaScript, a module is a JavaScript file that encapsulates a specific set of functionalities or resources. A module file contains code that defines variables, functions, classes, or objects that can be exported and imported by other JavaScript files.

-------------------------------------------------End of 3rd Day training--------------------------------------------------------

DAY 4 : Import react and reactdom in our project

Html Code

<html>

<head>

<title>Page Title</title>

<script type="module" src="app.js"></script>

</head>

<body>

<div id="root"></div>

</body>

</html>

Js Code

import React from 'react';

import ReactDOM from 'react-dom';

const heading1 = React.createElement(

'h1',

{ id: 'headingFirst', className: 'headingFirstClass' },

'Hello, This is first heading');

const heading2 = React.createElement(

'h1',

{ id: 'headingSecond', className: 'headingSecondClass' },

'Hello, This is Second heading');

const mydv = React.createElement(

'div',

{ id: 'headingThird', className: 'headingThirdClass' },

[heading1,heading2]);

const root=ReactDOM.createRoot(document.getElementById("root"));

root.render(mydv);

In the above JS code,

1. Command: import React from 'react';

Use: this command is use to import react in our js file.

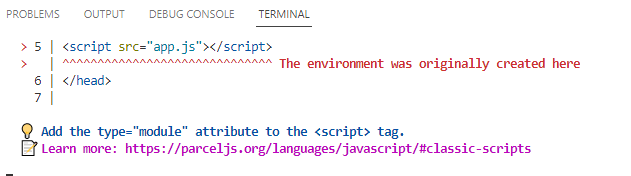
1. Command: import ReactDOM from 'react-dom';

Use: this command is use to import react-dom in our js file.

In the above html code,

<script type="module" src="app.js"></script>

we have to include script tag in head tag, and the type of this script tag should be module always, if we not use type as module than we get this error



---------------------------------------------------------------End of 4th Day training-------------------------------------------

DAY 5 : Use of JSX

Q.1 What is JSX?

JSX (JavaScript XML) is an extension to the JavaScript language syntax used by React and other libraries/frameworks to define the structure and appearance of user interfaces. It allows developers to write HTML-like code directly within JavaScript, combining the power of JavaScript with the expressive nature of HTML-like syntax.

JSX looks similar to HTML, but it is not valid HTML. It is a syntactic sugar provided by React to create React elements (virtual representations of DOM elements) in a more intuitive and declarative way. JSX allows you to describe the structure of UI components and specify how they should render based on the data.

Q.2 What is a functional component in React?

**ReactJS Functional components**are some of the more common components that will come across while working in React. These are simply JavaScript functions. We can create a functional component in React by writing a JavaScript function. These functions may or may not receive data as parameters. In the functional Components, the return value is the JSX code to render to the DOM tree.

Different ways of calling the functional component:

We can call the functions in javascript in other ways as follows:

**1. Call the function by using the name of the function followed by the Parentheses.**

// Example of Calling the function with function name followed by Parentheses

import React from 'react';  
import ReactDOM from 'react-dom/client';  
function Parentheses() {  
 return (<h1>  
 We can call function using name of the   
 function followed by Parentheses   
 </h1>);  
}  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(Parentheses());

**2. Call the function by using the functional component method.**

// Example of Calling the function using component call

import React from 'react';  
import ReactDOM from 'react-dom/client';  
function Comp() {  
 return (<h1> As usual we can call the function using component call</h1>);  
}  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Comp />);

Now, We will use the functional component method to create a program and see how functional components render the component in the browser.

Program to demonstrate the creation of functional components

Open your React project directory and go to the *src*folder then edit the **index.js** file.

Javascript

|  |
| --- |
| //index.js File  import React from 'react';  import ReactDOM from 'react-dom';  import Demo from './App';    ReactDOM.render(      <React.StrictMode>          <Demo />      </React.StrictMode>,      document.getElementById('root')  ); |

Open the *App.js* file from the *src*folder and edit the file.

Javascript

|  |
| --- |
| //App.js File  import React from 'react';  import ReactDOM from 'react-dom';    const Demo=()=>{return <h1>Welcome to React</h1>};  export default Demo; |

JS Code

import React from 'react';

import ReactDOM from 'react-dom';

const heading1 = React.createElement(

'h1',

{ id: 'headingFirst', className: 'headingFirstClass' },

'Hello, This is first heading');

const heading2 = React.createElement(

'h1',

{ id: 'headingSecond', className: 'headingSecondClass' },

'Hello, This is Second heading');

const mydv = React.createElement(

'div',

{ id: 'headingThird', className: 'headingThirdClass' },

[heading1,heading2]);

const Mydv2 = () => {

return(

<>

<div id="div1" key="div1">DIV1</div>

<div id="div2" key="div2">DIV2</div>

<div id="div3" key="div3">DIV3</div>

</>

);

};

const root=ReactDOM.createRoot(document.getElementById("root"));

root.render(<Mydv2 />);

In the above code ,

const Mydv2 = () => {

return(

<>

<div id="div1" key="div1">DIV1</div>

<div id="div2" key="div2">DIV2</div>

<div id="div3" key="div3">DIV3</div>

</>

);

};

Above code is react functional component

-------------------------------------------------------------End of 5th day training---------------------------------------------

DAY 6 : Babble, Virtual DOM, React Fiber

Q.1 What is babel?

Babel is a very famous transpiler (Compiling tool in simple words) that basically allows us to use future JavaScript in today’s browsers. In simple words, it can convert the latest version of JavaScript code into the one that the browser understands. The latest standard version which JavaScript follows is ES2020 which is not fully supported by all browsers hence we make use of a tool such as ‘babel’ so that we can convert it into the code that today’s browser understands.

Q.2 What is a transpiler?

It is a tool that is used to convert source code into another source code that is of the same level. That is why it is also known as a source-to-source compiler. Both codes are equivalent in nature, considering the fact that one works with the specific version of the browser and one doesn’t.

Q.3 Why do we need Babel?

The main reason we need Babel is that it gives us the privilege to make use of the latest things JavaScript has to offer without worrying about whether it will work in the browser or not. Simple Example

Before installing and making use of all the features of the Babel tool, let’s see a simple code of the latest standard version of ES2017 and see what happens to it when we pass it into the Babel engine.

Q.4 What is Virtual DOM?

React uses Virtual DOM exists which is like a lightweight copy of the actual DOM(a virtual representation of the DOM). So for every object that exists in the original DOM, there is an object for that in React Virtual DOM. It is exactly the same, but it does not have the power to directly change the layout of the document. Manipulating DOM is slow, but manipulating Virtual DOM is fast as nothing gets drawn on the screen. So each time there is a change in the state of our application, the virtual DOM gets updated first instead of the real DOM. You may still wonder, “Aren’t we doing the same thing again and doubling our work? How can this be faster?” Read below to understand how things will be faster using virtual DOM.

Q.5 What is “React Fiber”?

Fiber is the new reconciliation engine in React 16 (Works on def Algo). Its main goal is to enable

incremental rendering of the virtual DOM.

Q.6 Command to prepare production Build?

Npx parcel build reachHello.html (npx is use for manually run)

---------------------------------------------------------------End of 6th Day training-------------------------------------------

DAY 7 :