Namaste React By Akshay Saini

Episode - 01 Inception

VScode know Use something known as Emmit

Emmit - Generate some code for us.
E.g html:5 → Gives us basic HTML skeleton.

Browsers have the Javascript Engine in it, that executes these Javascript.

Note: Browser do not understand react code

To use react in our project there is 2 way

1] Using CDN →

CDN → Content Delivery Networks.
websites where react is hosted and pulling react into our project

CDN is a place where react Library Is hosted.

Question to search → what is CDN?
what is crossorigin In CDN Link?

React is built using Javascript.
↳ It is javascript Library
↳ Developed by facebook developers

react.development.js → - Core file of react
- core react framework algorithm.

react-dom.development.js → - useful for dom operations
- modify the DOM.

Note: why there is separate 2 files for react ?
⇒ react is not only used for browser. It is also works or used on different places like mobile, desktop etc
for example,
React native for mobile applications

* First React Program -

1) Creating h1 tag into react -

↳ we create using React.createElement()
↳ It takes 3 arguments
- what type of tag need to create
- an object
- text to put or content

eg: const heading = React.creatElement("h1", {}, "Hello world from react");

2) Put this heading into div with id root

↳ firstly, we need to tell react what is the root where you need to render h1.

↳ React needs to have a root where it will do all the DOM manipulation.

↳ creating root in react

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

Note: creating any type of element it is the job of react or core react

↳ But creating a root or rendering into inside it, it is the job of reactDOM.

3) Rendering heading tag inside root

```
root.render(heading);
```

↳ root is the place where all the react code will run.

```
React.createElement("h1", {}, "text");
```

{} It is an object where we give an attribute to the tag.
eg id, classes, any other

```
const heading = React.createElement(
                "h1",
                { id: "heading" },
                "hello world from react"
            );

console.log(heading); //object
```

Note: The elements we create in react is an object not a tags.

Above example h1 type of object is created It is not a html tag.

It is react element. and all react elements are created as an object.

"React element is javascript object"

React elements have something known as <u>props</u>.

props → props are children + attributes that we pass in to element

2] root.render(heading);
↳ takes react element and convert it into HTML tag that browser will understand.

Nested Elements In React -

HTML structure is not a single tag structure
It is nested structure like

```
<div id="parent">
    <div id="child">
        <h1> I am h1 tag </h1>
    </div>
</div>
```

- Creating above nested HTML structure using react.

```
const parent = React.createElement("div",
    {id: "parent"},
    React.createElement(
        "div",
        {id: "child"},
        React.createElement("h1", {id: "h1"},
            "I am h1 tag")
    ));
```

```
React.createElement("h1", {}, "I am h1");
```

- "h1" → HTML tag need to create
- {} → props (attributes)
- "I am h1" → props (children) Go Inside html tag

- Creating Siblings (multiple elements) inside HTML structure

```
<div id = "parent">
    <div id = "child">
      <h1> I am h1 tag </h1>
      <h2> I am h2 tag </h2>
    </div>
</div>
```

If we have to pass more than on children to the react element we have to use array of children.

Create an array of childrens to pass multiple children

```
const parent = React.createElement(
    "div", {id : "parent"},
 React.createElement("div", {id: "child"},
[React.createElement("h1", {}, "I am h1"),
 React.createElement("h2", {}, "I am h2"),
])
);
```

This is core fundamental of react

Need of JSX -

- Creating HTML structure using React

```
<div id = "parent">
    <div id = "children">
        <h1> I am h1 tag </h1>
        <h2> Iam h2 tag </h2>
    </div>
</div>
```

```
<div id = "children"><h1> I am h1 tag </h1>
<h2> Iam h2 tag </h2> </div> </div>
```

```
const parent = React.createElement ("div",
    {id : "parent"},
[React.createElement ("div", {id: "child"},
[React.createElement ("h1", {}, "I am h1 tag"),
React.createElement("h2", {}, "I am h2 tag")]
),
React.createElement ("div", {id: "child"},
[React.createElement ("h1", {}, "I am h1 tag"),
React.createElement ("h2", {}, "I am h2 tag")
]),
]);
```

Creating complex structure of HTML using react or core react fundamentals is difficult. and also not able for understanding.

This is why there is something known as JSX

Note: when we have to create tags using react faster and more efficient way, we use JSX for it.
↳ mordern react development

Index.html

```
<div id = "root">
    <p> Aditi is here! </p>
</div>
```

app.js

```
root.render(parent);
```

this root.render method replace everything inside div with id "root" by the elements or react elements we are passing in it.

Here, P tag (Aditi is here!) is replace by react element we created ie parent while rendering.

How it works

↳ firstly, when browser reads HTML code It will see 'Aditi is here' on the code and print on browser

↳ as will as it see script, it will load react and reactDOM

↳ load app.js, and it will start executing code line by line

↳ It comes ~~react~~ root.render(parent).

It render parent element inside div with id 'root'.

Replaced not appended.

Here react.Js only working in div 'id: root'.
Because,
- ↳ we define root as div with id "root"
- ↳ every thing will render only inside this root element
- ↳ all html code works the same

why react call Library →

- ↳ react can apply to the ~~po~~ small portion of page itself.
- ↳ react only works in the place ~~where~~ whatever the place we make root as.

React Advantages

- ↳ we can also use react in existing apps as well
- ↳ include it in specific or small portion of our app.

Note: we can not create multiple root or rendering element in react