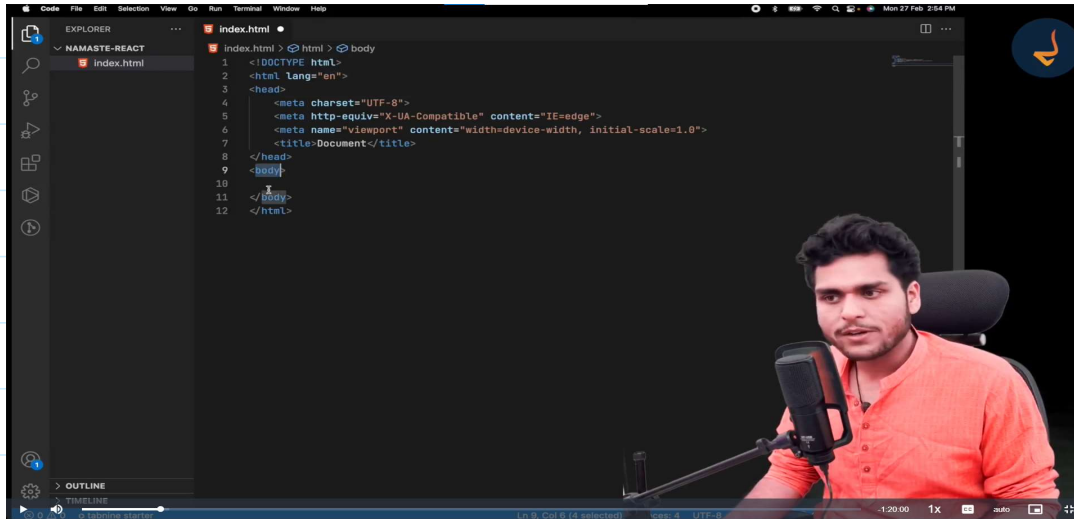


Step1: EMMET

html:5



Code Snippet 1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Namaste React</title>
</head>
<body>
  <div id="root">
    <h1>Hello Workd!</h1>
  </div>
</body>
</html>
```

Code Snippet 2:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Namaste React</title>
</head>
<body>
  <div id="root">

  </div>
  <script>
    const header = document.createElement("h1");
    header.innerText = 'Hello World from Javascript';
    document.getElementById('root').appendChild(header);
  </script>

</body>
</html>
```

Homework

1. What is CDN?
2. Why do we use CDN?
3. What is Cross Origin

Ways of adding React into our project:

1. Via CDN

```
<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>
```

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

There are two files one is react.development.js & second one is react-dom.development.js

react.development.js is the core react library.

react-dom.development.js is for manipulating DOM.

React is not only for web applications.

React is also used for making mobile applications, 3D animations.

for any of these react.development is core react library, based on web/mobile application we will use react-dom/react native.

Code Snippet 3:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Namaste React</title>
</head>
<body>
  <div id="root">

  </div>

<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>
<script>
  const heading = React.createElement("h1", {}, "Hello World From React");
  const root = ReactDOM.createRoot(document.getElementById("root"));
  root.render(heading);
</script>
</body>
</html>
```

Code Snippet4:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Namaste React</title>
  <link rel="stylesheet" href="./index.css">
</head>
<body>
  <div id="root">

  </div>

<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>
<script src="./app.js"></script>
</body>
</html>
```

Attributes

```
//app.js
const heading = React.createElement("h1", {id: 'heading'}, "Hello World From React");
console.log(heading);
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(heading);
```

```
#heading{
  color: red;
}
```

Output:

Hello World From React

React.createElement(); will return a react element.

React element is nothing but a object.

```
{
  "type": "h1",
  "key": null,
  "ref": null,
  "props": {
    "id": "heading",
    "children": "Hello World From React"
  },
  "_owner": null,
  "_store": {}
}
```

This object is then passed to render method, which actually converts to html element

And rendered to DOM.

How to create nexted elements?

Task1:

```
<div id="parent">
  <div id="child">
    <h1>I'm a h1 tag</h1>
  </div>
</div>
```

Solution1

```
//app.js
const heading = React.createElement("div", {id: 'parent'}, React.createElement("div",
{id: 'child'}, React.createElement("h1", {}, "I'm a h1 tag")));
console.log(heading);
const root = ReactDOM.createRoot(document.getElementById("root"));
console.log(root.render(heading));
root.render(heading);
```

React.createElement takes three arguments

- 1) Type of Element
- 2) Attributes
- 3) children

How to create nested elements? Alongwith multiple child elements

Task2:

```
<div id="parent">
  <div id="child">
    <h1>I'm a h1 tag</h1>
    <h2>I'm a h2 tag</h2>
  </div>
</div>
```

Solution2:

```
//app.js
const heading = React.createElement(
  "div",
  { id: "parent" },
  React.createElement("div", { id: "child" }, [React.createElement("h1", {}, "I'm a
h1 tag"), React.createElement("h2", {}, "I'm a h2 tag")])
);
console.log(heading);
const root = ReactDOM.createRoot(document.getElementById("root"));
console.log(root.render(heading));
root.render(heading);
```

We will pass multiple children using array.