

INCEPTION

Hello world using HTML

Simply type html:5 in vs code it will generate basic html code with header and body by making use of emmet

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div id="root">
    <h1>Hello world</h1>
  </div>
</body>
</html>
```

Hello world from JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div id="root">
    </div>

  <script>
    let heading = document.createElement('h1');
    heading.innerHTML = 'Hello world form JavaScript';

    let root = document.getElementById('root');

    root.appendChild(heading);
  </script>
</body>
</html>
```

Injecting React superpowers using CDN

- We have to add two CDN links one is for React core (react.development.js) another one is for
- DOM operations (react-dom.development.js).
- React have two CDN because the core react we will be able to use with other react based apps such as React Native, React 3D etc.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="./index.css">
  <title>Document</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>
</body>
</html>
```

Hello world from React

```
<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>
    let heading = React.createElement('h1', {id: 'heading'}, 'Hello world
from React');

    let root = ReactDOM.createRoot(document.getElementById('root'));

    root.render(heading);
  </script>
</body>
```

- The createElement() method of React is having three arguments they are Tag, Attributes and children respectively
- createElement(<tag>, <attributes>, <children> or [<child1>, <child2>])
- createElement() will return an object which is having props, which is a combination of attributes and children
- Root is where all our react elements will get rendered
- All the contents we written inside the root elements will get replaced by the content which we will add with root.render() method
- We will be able to add React to an existing JQuery applications without affecting anything as it is a library
- Order in which the script imports do matter as well as the 'root' where we render the React, that can be small portion of the page as well

Complex structure implementation ex:

```
* <div id="parent">
*   <div id="child">
*     <h1>I'm an h1 tag :) </h1>
*     <h2>I'm an h2 tag :) </h2>
*   </div>
*   <div id="child2">
*     <h1>I'm an h1 tag :) </h1>
*     <h2>I'm an h2 tag :) </h2>
*   </div>
* </div>
*/

let parent = React.createElement('div', { id: 'parent'}, [
  React.createElement('div', { id: 'child'},[
    React.createElement('h1', {}, "I'm an h1 tag"),
    React.createElement('h2', {}, "I'm an h2 tag")
  ]),
  React.createElement('div', { id: 'child2'},[
    React.createElement('h1', {}, "I'm an h1 tag"),
    React.createElement('h2', {}, "I'm an h2 tag")
  ])
]);

let root = ReactDOM.createRoot(document.getElementById('root'));

root.render(parent);
```