Namaste React

Ep 1) Inception

- **Emmet Abbreviation**: It will generate default some lines of code, for you to speedup the process of writing code by using shortcut

- Browser doesn't understand React. Browser understand only Javascript:

- Adding React to your project
  * **cdn React**   [cdn- content delivery network, These are the website where, React has been hosted & we are just pulling React using cdn links]
    ↳ cdn is a place where these React library are hosted and are using in our code using cdn links

  cdn links:- react.development.js → used to Import Core React library in our code
  react-dom.development.js → used to Import React dom in code

---

Hello World using HTML

```
<html>
  <head>
    <title>HTML</title>
  </head>
  <body>
    <div id="root">
      <h1>Hello World</h1>
    </div>
  </body>
</html>
```

Hello World using Js

```
<html>
  <head>
    <title>Js</title>
  </head>
  <body>
    <div id="root"></div>

    <script>

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

    head.innerHTML = "Hello World";

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

    root.appendchild (head);
    </script>

  </body>
</html>
```

Hello World using React

```
<html>
  <head>
    <title>React</title>
  </head>
  <body>
    <div id="root"></div>

    cdn links:-

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

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

    <script>
    const heading = React.createElement("h1", {}, "....");
    const root = document.getElementById ("root");
    const root1 = ReactDOM.createRoot (root);
    root1.render (heading);
    </script>
  </body>
</html>
```

React.createElement (createElement belongs to React core
ReactDom.createRoot (createRoot belongs to React-dom)

React Works with the Concept of Virtual Dom:

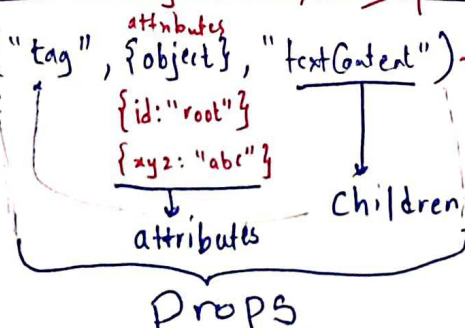- Means whenever you click on browser instead of updating whole Dom TREE only a particular (node/element) in TREE is updated.
  - Which increases the performance / optimization of Website.
  - Virtual Dom - Instead Refresing the whole webpage on every click, only particular element is updated.

React.createElement ("tag", {object}, "textContent") → 3 parameter
                                    attributes
  ↳ object (Js object):   {id:"root"}
     (React Element)   {xyz: "abc"}
                                    ↓
                              attributes   Children
                                    Props

```
<div id."parent">
  - <div id."child1">
       <h1> h1 Tag </h1>
       <h2> h2 Tag </h2>
    </div>
  - <div id."child2">
       <h1> h1 Tag </h1>
       <h2> h2 Tag </h2>
    </div>
</div>
```

```
const parent: React.CreateElement("div", {id:parent"},
[ React.create Element ("div", {id : "child1"},          39
   [ React.createElement ("h1", {}, "I am h1"),
     React.createElement ("h2", {}, "I am h2"),
   ]);
  React.create Element ("div", {id :"child2"}, [
    React.createElement("h1" {}, "h1 tag")
    React.createElement ("h2", {}, "h2 tag")
  ]);
```

Nested HTMl Tag in HTML   Nested HTMl Tag in React

But writting these code in React is Complex. Hence JSX comes into a concept.

- Note Imp: If your 'root' element in body tag already contains some other tag, then React will overwrite everything inside "root" and replace with whatever given inside render.