React JS is a library and not a framework

* library is a collection of prewritten function

Why React?

- Component based Approach (targets single component)
- Uses a declarative Approach
- DOM updates are handled gracefully (eg: Youtube)
- Reusable Code
- learn once, write any where
- designed for speed, speed of implementing the application and scalability

#Babel is a transpiler (a library) to convert JSX to pure JavaScript and latest version of JS to older version which is supported everywhere

```
Input: ES2015 arrow function
[1,2,3].map(n => n+1);

Babel Output: ES5 equivalent
[1,2,3].map(function(n) {
    return n+1;
});
```

eg of JSX: const name = <h1> Mohit Kumawat </h1>;
element

const name = "Anurag"; Pure JS
 Anurag ; HTML Element

Const name = Anurag ; JSX Element

* Whenever we create React Project there will be one and only one index.html and other will be JSX file

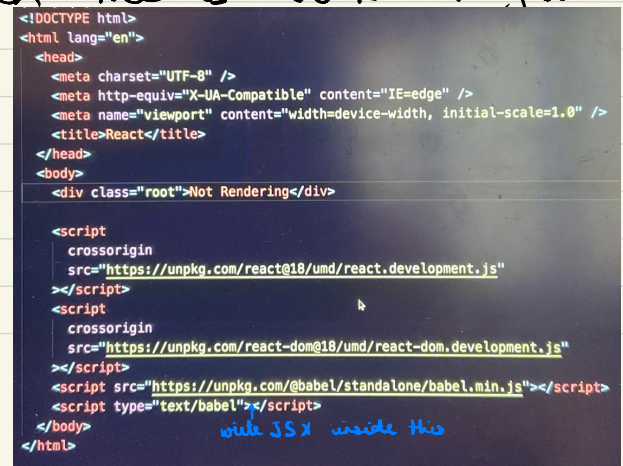
JSX Element
*Const header =
<header>
<h1> Hello </h1>
 Hello Namaste
</header>

} one Component (one tag)

* include in script both react and react-DOM
↓ this is for overall react
for eg of Youtube

react-DOM is inside react

* crossorigin means work everywhere



```
<script type="text/babel">
const element = <h1>Hello</h1>;
const rootelement = document.querySelector(".root");
ReactDOM.render(element, rootelement);
```

ReactDOM.render(What I have to do, where);

```
<html lang="en">
  <head>
    <title>React</title>
  </head>
  <body>
    <div class="root">Not Rendering</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="https://unpkg.com/@babel/standalone/babel.min.js"></script>
    <script type="text/babel">
      const rootElement = document.querySelector(".root");
      const element = <h1>Hello FSJS 2</h1>;
      ReactDOM.render(element, where);
    </script>
  </body>
</html>
```

To target here in HTML

I have declared this style but not used it

+always camelcasing

```
const CSS={border: "2px solid black",backgroundColor: "yellow"};
const header= (
  <header>
    <h1> Hello, Welcome Back </h1>
  </header>
);
const main = (
<main style={CSS}>
  <p> Lorem </p>
  <p> Lorem </p>
</main>
);
```

JSX

JSX

here used the object which I created

```
const footer = (
  <footer>
    <p> @ 2002 </p>
  </footer>
);
```

JSX

To include JSX element inside another Element we use ()

```
const app = (
<div>
  {header}{main}{footer};
</div>
);
ReactDom.render(app, rootElement);
```

(will work also without comma)

* as we can see div does not make sense above
So hence **React Fragment** came into Picture

```
const app = (
<>
  {header},{main},{main},{main},{footer};
</>
);
```

+ Reusable code

* in header

```
const header = (
<header style={{border: "2px solid black", color: "red"}}>
```

key value pair

double curly brackets because & are creating an Object

* in Javascript we write className ✓ & not class ✗

[as class is already reserved in Javascript]

```
<style>
  .header {
    border: 2px solid black;
  }
</style>

const header = (
<header className="header">
  <h1> Hello </h1>
  <p> Welcome </p>
</header>
);
```

(Takes more space)

npm : whatever we do will be done globally

npx : at specific Place

* in terminal

FSJS2 : npx create-react-app namaste

Happy Hacking!

cd namaste

npm start

npm-modules : holds all the necessary packages for react

* disable HTML Plugin

* manifest.json file is used for progressive web-app (installing particular website on Phone)
  eg: Youtube, Twitter

* robots.txt : Web scraping ?
  Whatever content we write how is it ranked, SEO and all

* Src : here we write all the code

* package.json : whatever we install will be noted here

* package-lock.json : whatever we write will be locked here

* gitignore : files we want to ignore

* components names will start with capital letter
  Header
  Main
  Footer