# REACT

- JS Library created by facebook.
- Used for creating UI (User-Interface)
- let's you devide complex UI into small/isolated pieces of code called Components.
- components :- Independent & reusable piece of code. Same as JS functions but work in isolation and return HTML.

  1. ~~Compone~~ Class Based
  2. ~~Com~~ Function Based

# WHY REACT?

- JS is based on imperative approach (step by step)

  ↓

  Fetch An element,
  style it,
  fill content,
  append it.

But In React we just need to tell END state & React'll do it by itself.

React is Based on Declarative Approach. So, we need to write less code.

- We can create web App Quickly & efficiently ~~with~~ than vanilla JS.
- React follows ~~SA~~ SPA approach.

Single Page Application - SPA

Load Single HTML file & modify it dynamically.

## React Alternatives:-

↳ Angular
↳ Vue

- Readability ✓
- DRY (Don't Repeat Yourself) principle ✓
- Reusability ✓
- SOC (seperation of Conceon) ✓
- Maintainability ✓

modular Code - Divide the Code into seperate Components.

We can say Components are Custom HTML Elements which we can import in index.js & then use it.

# SETUP REACT

WAY ⇒ 1.)

Step-1.) Install VS CODE & nodeJS

Step-2.) Create a new folder → ReactFolio

Step-3.) Change directory to → ReactFolio

Step-4.) npx create-react-app projectname

Step-5.) Change Directory to → projectname

Step-6.) npm start or npm run start

WAY ⇒ 2.) Use Replit (online website for creating projects)

WAY ⇒ 3.) Clone react-tailwind-css-starter-pack Repository from Github.

package.json – has project Dependencies, scripts and all info about the Project.

Src > index.js – Entry point in React Application. it calls ReactDOM.render()

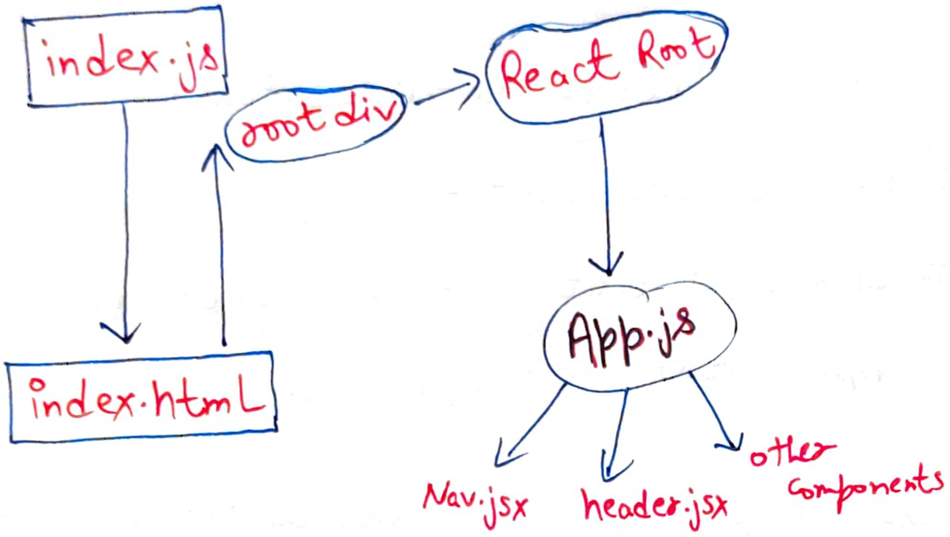
Src > App.js – Root Component

node-modules – has all the Required Packages.

public > index.html – Single & Main HTML file.

⇒ We Never Share node-modules/ upload folder Bcz it's size is very large. It can be easily created using a command.

~~npm start~~ npm install



## JSX

JavaScript XML

Syntax Extension To JS.

Allow us to write HTML Code in JS.

```
const name = "Himanshu"
<h1> User : {name} </h1>
```

Src > Components> UserData.jsx

```
export function UserData (props){
        return (
            <div>
                <h1>{props.name}</h1>
                <p>{props.day}-{props.month}-
                                {props.year}</p>
            </div>);
}
```

Src > App.jsx

```
    import user from `./component/UserData';
    export function App(){
        const response=[ Array of objects
                {name: "Himanshu",
                 day : 1,
                 month : 11,
                 year : 1999},
                {
                          }
            ];
    return (
        <User name={response[0].name}
              day = {response[0].day}>
        </User>);
}
```

Props :- ⊙ properties

⊙ props are the arguments passed into React Components.

⊙ are like func. arguments in JS. & attributes in HTML

☆ If you put content inside opening & closing tag of component invoking, like this.

```
<Card>
  <User ——> </>
  Hello world
  <User > —— </>
</Card>
```

It will not be displayed on the Browser. But we can do it with. props.children

```
function Card (props) {
        ~~<div>~~
        return(
            <div>
                {props.children}
            </div> );
}
```

★ Props are read-only objects containing values of attributes

## Concepts Learned :-

1. Core Syntax
2. Setup React
3. JSX
4. Component
5. Props - Deal with Data
6. props.children
7. props.className - H.W

- Props in React is used to pass Data from one component to another (Parent component to child components.