**REACT**

* React ?
* React is a open-source, dynamic library….
* We can reuse components in React most of the time
* Supports JSX, SPA, Virtual DOM
* Angular ?
* It’s a open-source, dynamic framework
* Components are not reusable most of the time.
* For creating large applications it’s mostly used
* Supports routing and many more stuff in Angular.
* Vue?
* Vue is a open-source, dynamic framework
* Functions ready to use
* Supports Event Handling.
* Supports routing.

REACT have some features like mentioned:

1. JSX=>

It’s a combination of JS and HTML code

1. Virtual DOM=>

It’s a copy. For example there is ‘A, B, C, D’ parts and there is a update in C & D part. Real DOM will make change in whole ‘A, B, C, D’ but Virtual DOM compares it with the real DOM and see that there is a change in only C & D part and so it will make update in C & D part only.

i.e,

1. In JS the whole DOM part is updated but
2. In REACT it finds the difference between Real and Virtual DOM and update what is needed to update.
3. SPA(Single Page Application)=>

No more web-pages just a web-page i.e single page. Same page will contain multiple elements. It is light weight because of SPA.

Library and Framework Difference

* Library=>

Programmer decide how much you have to

use what you have to use as and when it is required it's full code. It gives us functions that we want from which we divide that what we have to use and when and where we have to use it and how much.

Example=>If we prefer library, we have a house and it have a balcony or two balconies and we have option here that we have to use a balcony of house or not alike framework.

* Framework=>

Its a half-written code it's not complete code that we use. It gives us a blueprint that we can use it's give us a structure. I have to insert or complete the structure that we required just like example it provides us a skeleton of something and the we have to follow compulsory/mandatory.

Example=> If we prefer framework, here also we have a house and it have one balcony and we don’t have any option we have to use it’s balcony compulsory.

* Component =>
* Anything that we make in front-end React is a component.
* Your Component should Start with Capital Letters Always.

|  |  |
| --- | --- |
| Function Based Component |  |
| 1. We use rfc in small letter for creating Function Based Component |  |
|  |  |

If we have to write more than one elements inside a return statement

For example:

function FuncCompo() {

    return (

<h1>Hello World</h1>

<p>Hello World</p>

    )

}

This code will give me error

We can write this h1 & p inside another tag<></> that is inside a parent tag

Example

function FuncCompo() {

    return (

<>{/\* This will not give error <> is a parent tag  \*/}

<h1>Hello World</h1>

<p>Hello World</p>

</>

    )

}

This code will not give me error

export VS import

|  |  |
| --- | --- |
| export | import |
| The component which we have to send it out side we write it using export | The component which we want to use in particular file or folder we write it with import |
| E.g : export default File\_name | E.g : import File\_name from './Path';  For getting the component we write this upper import statement  And for using / calling the component at particular place we can simply write  <Filename/> =><NewComp/> like this inside the tag   1. g:   import logo from './logo.svg';  import './App.css';  import NewComp from './NewComp';  function App() {    return (      <div className="App">       <h1>Welcome to EA20 Batch</h1>       <NewComp/>      </div>    );  }  export default App; |

There are two types of export

export default &