* Develop in React →

* Babel and Webpack Overview →

→ Babel is a library with a lot of code inside it, and it has some certain task to perform, out of which two tasks are -

(1) Coverts ES6 code to ES5.

(2) Converts JSX syntax.
[Javascript XML]

(1) → Since, a lot of browsers don't support ES6 so, before we go ahead and run our code it has to be converted to ES5, and babel does all the task of converting the code b. we have in our code base to ES5.

(2) → So, as we saw JSX before, that particular JSX gets compiled into react.create element.

* Webpack → It's a tool that lets you bundle your Javascript pages.

→ As in react we will having a lot of .js files already in the libraries we dowonloaded and apart from that we will be creating .js files on our own as well as per the requirement.

So, the webpack goes ahead and bundles up all the javascript files into a one file known as "bundle.js"

* More On Components →

→ After we have installed 'create-react-app' in our folder we will see a multiple different folders present inside it.

→ We have a 'public' folder which Contains our 'index.html' which is mandatory for our website.

→ We have a file named as 'manifest.json' which we can say is the meta-data of our website.

→ It provides us with the information about our website that is the name of the website, description of the website, apart from it when you are running the website on desktop or mobile it will give you an option add to home screen in order to access that website by directly clicking on the icon present on home screen, and for that we need logo which are also defined inside this file.

→ Then we have a file named as 'robots.txt' which is basically written for the search engines.

→ 'index.js' is the main file which our projects looks for when we hit npm start.

→ function Heading() (name of the components always starts with a capital letter)

{

return(<h1> Welcome to React </h1>)

}

Then we do,

```
ReactDOM.render(<Heading />,
                document.getElementById("root"));
```

* So, if we want to return multiple elements, and since return statement can only return one element, in order to achieve that we simply put all our elements inside a tag, so that when the return statement is executed it will only return the 'div'.

Ex →

```
function Head()
{
  return
  (
    <div>
    <h1> Hello </h1>
    <p> How are you </p>
    </div>

  )
}
```

So, what happens in the previous example is that an extra node for the element is also been created on the screen even if we don't need it, so in order to remove it we ~~have~~ can use →

```
{ <React.Fragment>
    <h1> ____ </h1>
    <p> ____ </p>
  </React.Fragment> }
```

or simply we can have empty tags.

```
{ <>
    <h1> ____ </h1>
    <p> ____ </p>
  </> }
```

This will stop the creation of an extra node in the DOM.

* In 'index.js' we are supposed to have the render statements and different '.js files' can be created for different components.

Then we need to import that component file in the index.js and export from component file.