Episode -3; laying-the-foundation

we will create scripts in package.json for npx parcel index.html and npx parcel build index.html

"scripts": {

"start": "parcel index.html ",

"build": "parcel build index.html",

"test": "jest"

},

now we can execute script in cli **npm run start (for start we can use npm start)** or **num run build** react element is equal to dom element and its just an Object

JSX:

JSX is a javascript syntax which is easier to create react element

Jsx is diffrenet than react , we can write react code without jsx also and JSX makes over developer life easy

const jsxHeading = <h1> Namste React using JSX</h1>;

Jsx is not an html inside javascript and its jsx is html like syntax and its not a valid javascript and JS engine not reading jsx and js engine understand the ES6. And parcel making jsx readable for javascript engines

Jsx code transpiled by parcel so browser get understand

Parcel gives the responsibilty of transpile to babel

Babel::

Babel is Javascript package and its javascript compiler and its not developer by facebook

JSX =>(Converts into) React.createElement => ReactElement.js => Html Element (render)

Babel is converting this JSX code into react element

React Component :  
  
- lots of the people saying everything thing in react is a component yes that is true , in webpage if we see button ,footer , header ,sideber, input box, searchbar that is component in react

what is a component ?  
 There are 2 type of component in react, first one is class based component and second one is functional based component   
  
 class based component : old way to writing code .now a days nobody using class based component,

functional based component : new way to writing code , it uses javascript function

what is react functional component ?   
Its a just a normal javascript function   
  
- whenever you are defining, functional component or any rect component name it in cpatital letter other wise you will get error as its react way to understand its a component  
  
const Headingcomponent =() =>{

return <h1> React Functional Component </h1>;

}

React Functional Component is function that will return some piece of JSX code. You can also say that React Functional Component is function that will return react element   
  
in above code I have used Javascript arrow function which is returning one line code so I can use it like const Headingcomponent =() => <h1> React Functional Component </h1>;   
  
if we want to wrap is in mutiple line then   
  
const Headingcomponent =() => (  
<h1> React Functional Component </h1>  
);   
  
So we use {} then we must need to write return  
  
we can aslo make it with nested in JSX as below   
  
const Headingcomponent =() => (

<div id =”container”>  
 <h1 class =”heading”> React Functional Component </h1>  
 </div>  
);   
  
  
if we want to render react component in root then we will write like   
  
root.render(<Headingcomponent / >);

component with angular brace for babel to understand its react component

----

Nested Component or composition of component  
  
  
if we want to add title compoent in above Headingcomponent then the code will be   
  
const Title =() => (

<h1 class =”title”> React With JSX and React Component </h1>  
   
);   
  
  
const Headingcomponent =() => (

<div id =”container”>

<Title />   
 <h1 class =”heading”> React Functional Component </h1>  
 </div>  
);   
  
As you can see we add Title component inside heading component that will work like we add html inside heading component   
  
 composition of component means component that include one or more component inside it.

In jsx inside curly braces you can run any piece of any java script code

const number = 1000;

const HeadingComponent = () => (

<div id ="container">

{number}

<h1 className="heading" >

Namste React Functional component!

</h1>

</div>

);

that will show number variable inside heading component, here you can make maths calucation in it also like {100 +200}  
  
const number = 1000;

const HeadingComponent = () => (

<div id ="container">

<h2>{number}</h2>

<h2>{100 +200}</h2>

<h1 className="heading" >

Namste React Functional component!

</h1>

</div>

);

you can inject any javascript inside curly parenthes and it comes as html in browser I mean you can also write console.log  
  
writing javascript inside JSX is very powerfull

What if I want to put react element inside react component I can do it in same way as component

const title = (

<h1 className="head" tabIndex="5">

{elm}

Namste React Using JSX !

</h1>

);

const number = 1000;

const HeadingComponent = () => (

<div id ="container">

{title}

<h1 className="heading" >

Namste React Functional component!

</h1>

</div>

);

you can put react element inside react element same way   
  
onst elm = <span>Element </span>

const title = (

<h1 className="head" tabIndex="5">

{elm}

Namste React Using JSX !

</h1>

);

and remember one thing we need to maintain the order I mean if we inject react element inside react element then we must need to define that element first otherwise we get error of undefined

---   
Cros Site Scripting:  
  
const data = api.getData();

const HeadingComponent = () => (

<div id ="container">

{data}

<h1 className="heading" >

Namste React Functional component!

</h1>

</div>

);

if we used api is bed api (attacked can attac )and that code executed and our app is running by many people and attcker get some access to user browser then that kind of attack known as cross site scripting   
  
here attacker an attack like it can steev cookies and read session storage, cookies and get many details of browser   
  
Jsx takes care of this kind of injection attacks even if api paases some melisies data into our code ,then JSX escape this   
  
whenever we write any code inside curly brasces jsx will senitize the code then pass it , it prevents cross site scripting so we dont need to take care of attacks feel free to use this kind of data as JSX is a such a powerfull thing