14 Feb 24

Scripts.

↳ We create a script to start our app using npm
So to do that we write in our

Pakage.json

"scripts" : {

"start" : "parcel index.html" ← this will start project in development mode.

"build" : "parcel build index.html" ← to build the project in development mode

✱

If you are new in the project go in the pakage.json and search for start or build to get started with any project. This are industry standards"

→ Now to start a app we will write

- npm run start ← to start or npm start
- npm run build ← to build

→ git push
→ ~~delete~~ comment all code from app.js except import from react

React Element / core of React

const heading = React.createElement("h1", {id:heading}, "Namaste React");

this creates a object

when we render to the dom it become a html tag.

const root = ReactDOM.createRoot(document.getElementById("root"));

↳ ReactDOM is used whenever we want oto display some item on screen.

root.render(heading);

when the root.render is used it will replace every thing that is present in that root

JSX

← It is not a part of React

→ It isa java script made by face book engineers to make it to write the react code.

JSX merges the HTML, CSS and javascript to make or write code easily.

⇒ `const jsxHeading = <h1> Namaste React </h1>;`

↳ It is not html inside Javascript

↳ It is similar to html

→ this creats a React object which then rendered by react and converted into Html tag

then `root.render(jsxHeading);`

So javascript does not understand this JSX directly it understands the Emmac script.

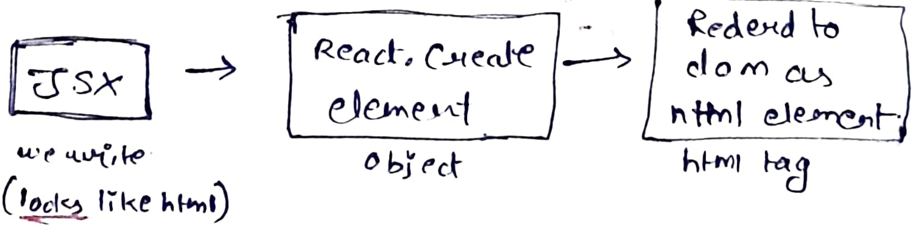
React → Parcel ↗ Babel helps us to write like this

This code is transpiles by the parcel before event it reaches the browser.

Babel → this helps to convert the code.

→ Java script compiler

→ One of the most important package.



→ In JSX we use Camel Case → tabIndex ✓

x tabindex x

Class

⇒ Const jsxHeading =<h1 className="head" tabIndex ="1"> Namaste React <h1>

↑ to create a class

↑ the element can navigated through keyboard tab key

In multiple line give parenthesis/Round brackets

```
const jsxHeading = (<h1 className = "head">
      Namaste React </h1>
)
```

Extensions
→
1) Prettier - format code
2) Brake pair colourizer
3) Better comments

#React Components

- ① Class based components
- ② Functional Components

Functional Component → A functino that returns a JSX code is a functional component.

first charecter is capital

```
Const  HeadingComponent = () => {
        return <h1> Namaste React Fun. Component </h1>;
    };
```

↑ return

or

```
Const HeadingCom = () => <h1> Name </h1> ;
```

or (Multiple lines)

bracket

```
const HeadingCom = () => (
        <h1> className ="heading"> Namaste React </h1>
);
```

Rendering a heading component ↓

```
root.render (<HeadingComponent/>);
```

Component Composition

```
const Title = () => <h1> This is title </h1>;
const Heading = () => (
(h1>
        <div id="container">
            <Title/>  ← Including one component into another component
            <h2> This is sub heading </h2>
        </div>
);
```

- We can write javascript inside your JSX code just using curly bases { }

```
ex →   const Heading = () => (
            <h2> This is subheading </h2>
            { console.log ("Hello"); }  ← Javascript
       )
```

Inserting ~~HTML inside~~ ~~a~~ ~~Javascript~~ React element inside component

```
ex →   const elem = <span> React Element </span>;
       const title = (
           <h1 className = "head" tabIndex="5 ">
               {elem} <br>
               Namaste JS !
           </h1>
       );

       Const Heading = () => (
           <div id= "container">
               {title}
               <h1> className = "head" >Namaste React Fun.Comp </h1>
           </div>
       );
```

Without running any script that is being passed inside the { } JSX Sanitizes it.

Which prevents Cross Site Scripting ← an type of attack that can be done by attacker,

-
 <br/> ⟺ <br/>

Same ~~m~~ meanings.