3. Laying the foundation

For execute using : npx parcel index.html
Instead of writing this command again and again we create/write script
Create script in package.json
Starting project in dev build script : `"start":"parcel index.html"`
Starting project in prod build script: `"build":"parcel build index.html",`

```
"scripts": {
  "start":"parcel index.html",
  "build":"parcel build index.html",
  "test": "jest"
},
```

So for executing dev we used
npm run start or **npm start**

Hello world in JSX

```
const jsxheading= <h1>Hello World JSX </h1>


const root =ReactDOM.createRoot(document.getElementById("root"));
root.render(jsxheading);
root.render(heading);
```

Javascript is code that js engine can understand
Js engine not understand JSX

JSX converted before it reaches the JS Engine it's converted by babel
Babel is Javascript compiler

How React code running ?

React.createElement => JS object (ReactElement) => HtmlElement (render)

How JSX code running ?

JSX => BAbel convert to React.createElement => JS object (ReactElement) => HtmlElement (render)

HTML	**JSX**
class	className
tabindex	tabIndex

In JSX single line

```
const jsxheading= <h1>Hello World JSX </h1>
```

No need to put in ()
But we can also put in () it's also fine

In JSX multiple line code wrap the code in round brackets ()

```
const jsxheading= (
                    <h1>Hello World JSX </h1>,
                    <h2>Hello World JSX </h2>
);
```

Functional component

Just normal javascript (arrow) function which return some piece of JSX code

First letter capital

```
// functional component

const HeadingComponent = ()=>
{
    return <h1>Functional Component</h1>
};
```

React element = JSX
So functional component also we can call it javascript function which return react element

```
const HeadingComponent = ()=>
(
    <div id="container">
     <h1>Functional Component</h1>
     </div>
);

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

root.render(HeadingComponent );
```

We can not render functional component like this because like this we rendering React Element

Need to render like this

```
//Functional Component
const HeadingComponent = () =>
(
    <div id="container ">
     <h1>Functional Component</h1>
    </div>
);
const root = ReactDOM.createRoot(document.getElementById("root"));

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

Component first Letter should be always in Capital

How to put component inside another component ?

```
const Title = ()=>

(
   <h1>React By Nikita</h1>
);


//Functional Component
const HeadingComponent = () =>
(

    <div id="container ">
      <Title/>
     <h1>Functional Component</h1>
    </div>

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

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

Put (render) one component inside another component known as Component composition

We composing two components into one

How to put react element in component ?

```
const Title =
  (
     <h1>React By Nikita</h1>
 );


//Functional Component
const HeadingComponent = () =>
(

   <div id="container ">
     {Title}
     <h1>Functional Component</h1>
    </div>


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

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

Here in react element title is by EOD normal javascript variable
So we will putting inside {}

How to put react element inside another react element ?

```
const subText =
(
    <h3>Creating Food Ordering App</h3>
)


const Title =
  (

     <h1>React By Nikita
        {subText}
     </h1>

 );
```

How to put component inside react element ?

```
//Component
const FooterComponent=()=>
(
   <div className="container">
       <h1>this is footer component</h1>
   </div>
);

//React Element
const Title =
(
    <h3>Creating Food Ordering App
        <FooterComponent/>
    </h3>
)


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

Anywhere in JSX in {} we can write Javascript expression (code)

JSX take care of malicious data and sanitising it
JSX making our code readable

Following syntax are same , exactly same working or same thing

```
<Title1/>
<Title1></Title1>
```

Here component is EOD javascript function

```
//Component
const TItle = () =>
{
  return <h2>Hello world JSX</h2>
};
```

so we can write also

```
{Title()}
```

And

This three are the same things

```
{Title()}
<Title/>
<Title></Title>
```