

## EPIISODE-03

### PART-01

We use Scripts Tags to start the project. Initially, we use **npx parcel index.html** instead of this we use **npx run start** or **npx start**.

**npx run build** is used for building the project.

we modify the script tags in package.json!

### PART-03

```
const heading=React.createElement("h1",{id:"heading"},"Learnig React Bro👋");  
const root=ReactDOM.createRoot(document.getElementById("root"));  
root.render(heading);
```

This Syntax is Really Tough and UnEasy To Understand!.

Then The **JSX(JavaScript Syntax)** is Entered.

JSX is Not HTML in JavaScript. JSX is Different from html.

But jsx is **HTML like Syntax**.

We Can Write React Code Without JSX Also. But It's a Complex one.

**JSX Syntax for h1:**

```
const jsxHeading=<h1>I am writing This From JSX </h1>
```

**Machine Cannot Understand This jsx? Then How ?**

**JSX(transplied before it reaches to JS Engine)-Parcel-Barbel**

**“bare” package is Converted The jsx code to Machine understandable Code.**

**How React Element Works?**

**React.createElement → React Element-JS Object → HTMLElement(render)**

**How JSX Works?**

**Jsx→Babel Converts This to → React.createElement → React Element-JS Object → HTMLElement(render)**

**That why Both Are Equal.**

Attributes of JSX is CamelCase Letters.

Explore Some of the attributes

## REACT COMPONENTS

There are Two Ways For Creating Components.

1)Class Based Components. – OLD

2)Functional Based Components. -NEW

➔What is Functional Based Components?

It is a javascript FunctionWhich return some piece of the jsx.

**Syntax:**

Method1:

```
const functionname= () => {  
}
```

Method2:

```
Const functionname=()=>();
```

We Render using Angular Brackets. (root.render(</>))

JSX will Take Care of the malicious data.

Three types of calling:

```
<Element10></Element10>  
  <Element10/>  
  {Element10()}
```