**Interesting Notes on React:**  
  
React components are JavaScript functions that return html markup  
  
Notice that <MyButton /> starts with a capital letter. That’s how you know it’s a React component. React component names must always start with a capital letter, while HTML tags must be lowercase.

//app.js  
//Source Code  
  
function MyButton() {

return (

<button>

I'm a button

</button>

);

}

export default function MyApp() {

return (

<div>

<h1>Welcome to my app</h1>

<MyButton />

</div>

);

}

A white rectangular sign with black text

Description automatically generated

**JSX** stands for JavaScript XML. It is a syntax extension for JavaScript often used in the context of React, a popular JavaScript library for building user interfaces. JSX allows developers to write HTML-like code within JavaScript, making it easier to define the structure and appearance of UI components.  
  
Many Tools to Convert Code into different Code:  
<https://transform.tools/html-to-jsx>  
React created At Facebook  
  
**Google Chrome Extension- Search “React Developer Tools”  
-Make sure to turn on allow access to file URLS for local files-Chrome, not sure on firefox**A screenshot of a web page

Description automatically generated

**It is common to use Visual Studio Code to program React.js.**<https://code.visualstudio.com/>

-Change Color Theme by going to Preferences/Color Theme/ “suggested to choose night owl”

**Code Sandbox- Free Editor In the Browser**Type “react.new” in browser  
  
**Also check out CodePen- to build and test out front end code.**[**https://codepen.io/**](https://codepen.io/)

**Also check out Glitch.com -Create next web project in browser with no setup.**[**https://glitch.com/**](https://glitch.com/)

**“CDN”- Stands for Content Delivery Network**

**import React, { Component } from 'react';**

**// Functional Component**

**function MyComponent() {**

**return <div>Hello, React!</div>;**

**}**

**// Class Component**

**class Counter extends Component {**

**constructor(props) {**

**super(props);**

**this.state = { count: 0 };**

**}**

**render() {**

**return <p>Count: {this.state.count}</p>;**

**}**

**}**

**// Props**

**function Welcome(props) {**

**return <h1>Hello, {props.name}</h1>;**

**}**

**// Event Handling**

**class Button extends Component {**

**handleClick() {**

**console.log('Button clicked!');**

**}**

**render() {**

**return <button onClick={this.handleClick}>Click me</button>;**

**}**

**}**

**// Conditional Rendering**

**function Greeting(props) {**

**return props.isLoggedIn ? <UserGreeting /> : <GuestGreeting />;**

**}**

**// Lists and Keys**

**function ListComponent(props) {**

**const items = props.items.map(item => <li key={item.id}>{item.name}</li>);**

**return <ul>{items}</ul>;**

**}**

**// Lifecycle Methods**

**class LifecycleExample extends Component {**

**componentDidMount() {**

**console.log('Component did mount');**

**}**

**render() {**

**return <div>Hello, React!</div>;**

**}**

**}**