

Understanding JSX | Complete React Course in Hindi #4

In the last tutorial, we refreshed our knowledge of JavaScript. In this tutorial, we will learn about JSX. Do remember to access the official [documentation of React](#).

In an early tutorial, we created our textutils project and discussed its folder structure. Let's continue our React journey.

React Components:

React has two types of Components:

1. **Function-based Component:** In short they are the JavaScript functions. Nowadays, Function-based components are primarily used.
2. **Class-based Component:** They are quite complex than the function-based component as the class components can work with each other.

Earlier we discussed, the code of app.js is displayed in our React app using index.js

Let's start our app using the command '*npm start*' in the terminal of "app.js". Your app will be served at *localhost:3000*

Tip: Firstly make sure both devices are connected with the same Wi-Fi. If you want to see your react app on your phone, then you can use the IP address provided in the terminal.

You can now view **textutils** in the browser.

Local: `http://localhost:3000`

On Your Network: `http://192.168.0.102:3000`

You might be thinking that HTML is written in 'return()' of app.js but it's actually JSX.

JSX:

It stands for Javascript XML. It is the syntax extension of Javascript which allows HTML to co-exist with Javascript in React.

Why JSX is created?

This feature makes sure that all the code stays unified to provide the developer with the best possible experience. It ensures that all files aren't scattered at different locations.

Some Points to remember while using JSX:

- You have to write "className" instead of "class" as class is the reserved keyword in Javascript
- In place of "for" you have to use "htmlFor"

JSX fragment feature:

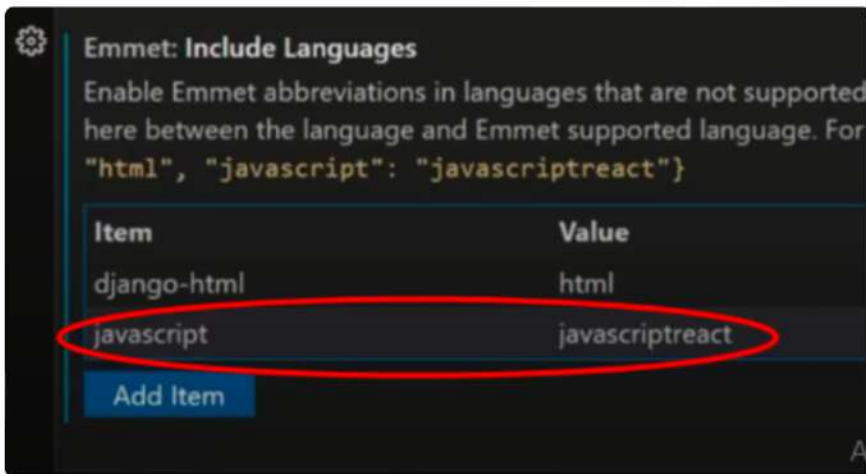
With `return()` we can return only one tag or Element. If you want to return two elements using JSX you have to use the JSX Fragment Feature or wrap the whole content in one element. For example, if we want to return the content of the `H1` tag and `Div` tag together we will be using a JSX fragment (`<>.....</>`).

Getting Started with our Own Application:

Earlier, We have created the app using the *create-react-app* package. But now we want to edit the default provided app and create our own application using React. So, Let's begin-
Firstly, remove the provided content of `return()` of `app.js`

```
function app(){  
  return(  
  
  );  
};
```

Tip: To use emmet abbreviation, Go to the settings and search for 'Emmet: include languages' and write javascript in the Item column and javascriptreact in the value column.



In app.js we will be creating a Navbar and paragraph component for our application and will be learning how to add CSS and Javascript.

```
function app() {  
  return (  
    <>  
    <nav>  
      <li>Home</li>  
      <li>About</li>  
      <li>Contact us</li>  
    </nav>  
    <div className="container">  
      <p>  
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
      </p>  
    </div>  
  </>  
)  
}
```

Adding CSS:

To add CSS to your application go to app.css available under Src Folder. Target your Element, and write your desired CSS code.

Adding JavaScript:

To add JavaScript in JSX we have to write the JS code in curly brackets.

```
1 | import './Appa.css';
2 |
3 | let name = "Harry3";
4 | function App() {
5 |   return (
6 |     <>
7 |       <nav>
8 |         <li>Home</li>
9 |         <li>About</li>
10 |        <li>Contact</li>
11 |      </nav>
12 |      <div className="container">
13 |        <h1>Hello {name}</h1>
14 |        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit
15 |          voluptatibus ut porro similique in, quod, molestiae commodi
16 |          mollitia?</p>
17 |      </div>
18 |    </>
19 |  );
20 | }
21 | export default App;
22 |
```

JavaScript variable, String

We will use curly Brackets to use JS. Here we are using the 'name' variable declared above.

Figure: Adding Javascript in JSX

A short summary of the tutorial:

```
1 | import './Appa.css';
2 |
3 | let name = "Harry3";
4 | function App() {
5 |   return (
6 |     <>
7 |       <nav>
8 |         <li>Home</li>
9 |         <li>About</li>
10 |        <li>Contact</li>
11 |      </nav>
12 |      <div className="container">
13 |        <h1>Hello {name}</h1>
14 |        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.
15 |          voluptatibus ut porro similique in, quod, molestiae commodi
16 |          mollitia?</p>
17 |      </div>
18 |    </>
19 |  );
20 | }
21 | export default App;
22 |
```

Importing CSS

Using Function based Component

JSX Fragment feature

Use className instead of Class

To use Javascript, We have to use Curly brackets.