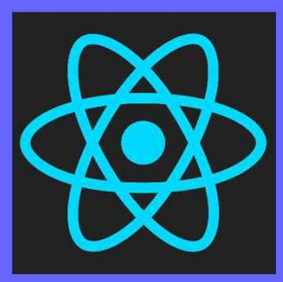
6

React JSX

Objectives

After completing this lesson, you should be able to do the following:

- Creating Components
- Basic Component
- Nesting Components
- Props



JSX

JSX

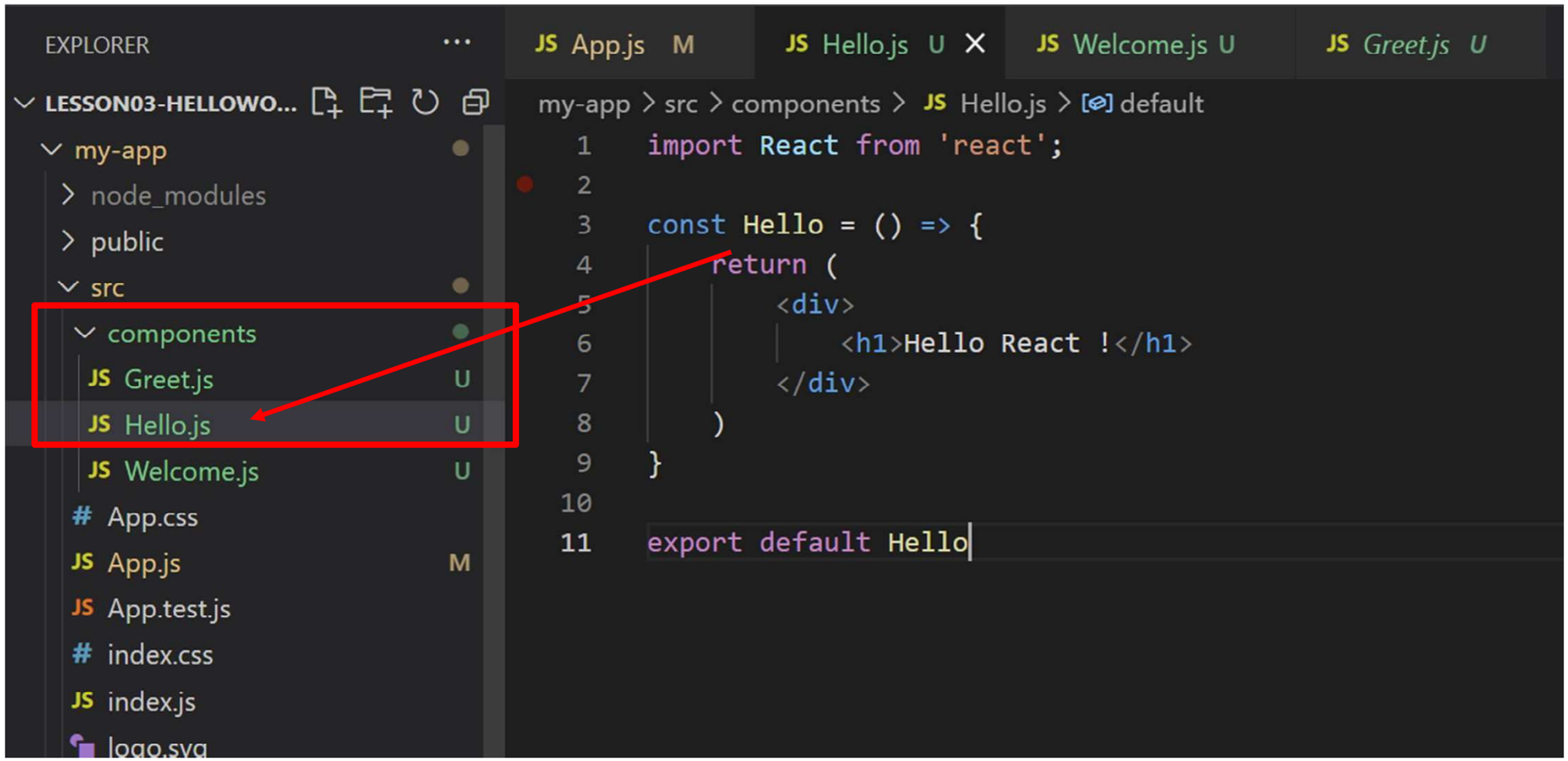
- JavaScript XML -- (Extension to the JavaScript Language Syntax)
- In React Lib Helps Writing XML – Like Code for Elements and Components
- Just Like XML [JSX Tags have a Tag Name, Attributes and Children]

- Is it Need to React Application
  - Ans is No [JSX is Not a Necessity to Write React Application]
- Why ?
  - JSX Makes your React Code Simpler and Elegant
- JSX Ultimately transpiles to Pure JavaScript which is Understood by the Browsers

- Lets see what does the code look like without JSX
- Let's create a react component using JSX and without using JSX
- That way we will not only understand how JSX translates to regular javascript but also appreciate how it brings out simplicity in your code

Lets Create Hello.js

- This Component will be Simple Functional Component which Render HelloMSG [Using JSX]

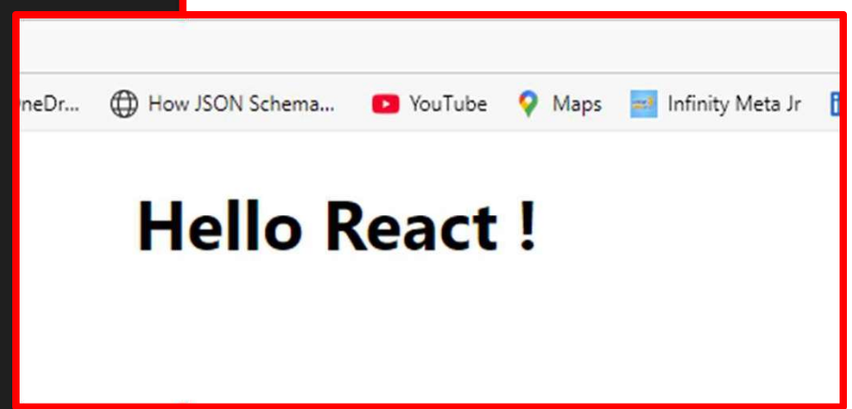


Include Component in App.js

JS App.js M × JS Hello.js U JS Welcome.js U JS Greet.js U

my-app > src > JS App.js > ...

```
1  import React, { Component } from 'react';
2  import logo from './logo.svg';
3  import './App.css';
4  import { Greet } from './components/Greet';
5  import Welcome from './components/Welcome';
6  import Hello from './components/Hello';
7
8  class App extends Component {
9    render() {
10     return (
11       <div className="App">
12         {/* <Greet />
13         <Welcome /> */}
14
15         <Hello />
16       </div>
17     );
18   }
19 }
20
```



Rewrite the Component Without Using JSX

- To Do So React Lib Provides createElement() Method

- Output is not as Expected, Since it has Considered <h1> as Simple Text Content

Solution

App.js M | Hello.js U | Welcome.js U | Greet.js U

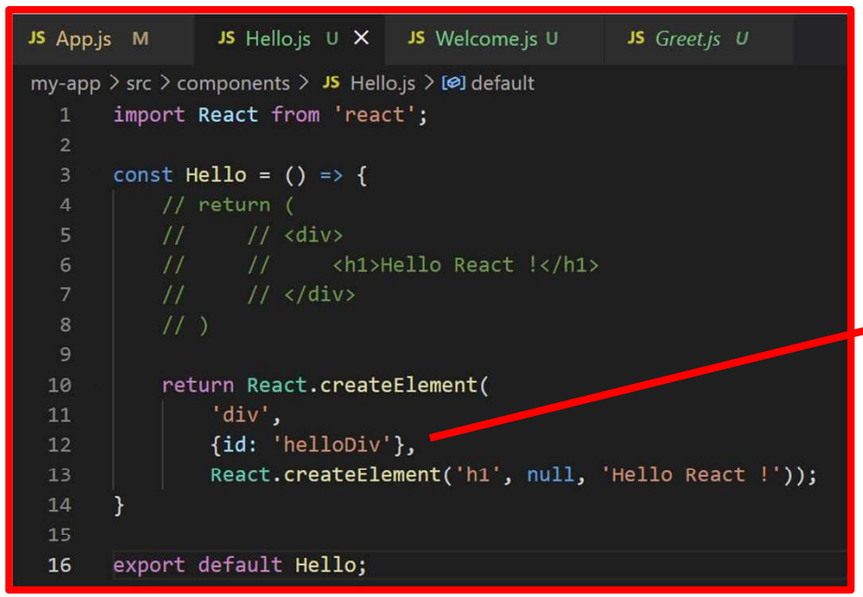
my-app > src > components > Hello.js > Hello

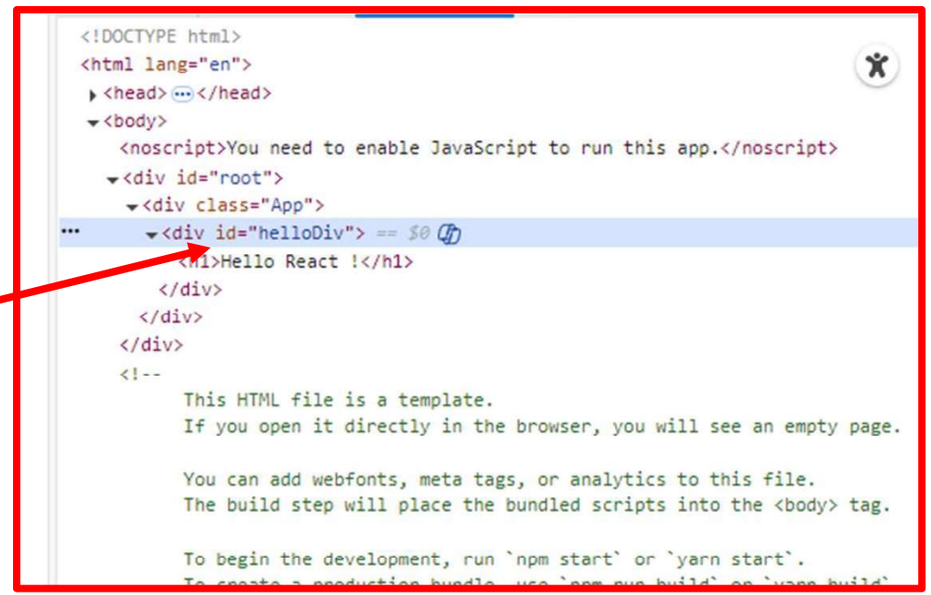
```
1   import React from 'react';
2
3   const Hello = () => {
4       // return (
5       //     // <div>
6       //     //     <h1>Hello React !</h1>
7       //     // </div>
8       // )
9
10      return React.createElement('div', null, React.createElement('h1', null, 'Hello React !'));
11  }
12
13  export default Hello;
```

Hello React !

Importance of 2nd Parameter in createElement()

- Second parameter basically is an object of key value pairs that will be applied to the element
- For example let's say we need an **ID attribute** on this **div tag** we can specify an object key is ID and the value is hello





App.js M | Hello.js U | Welcome.js U | Greet.js U

my-app > src > components > Hello.js > default

```
1  import React from 'react';
2
3  const Hello = () => {
4      // return (
5      //     // <div>
6      //     //     <h1>Hello React !</h1>
7      //     // </div>
8      // )
9
10     return React.createElement(
11         'div',
12         {id: 'helloDiv' , class: 'dummy'},
13         React.createElement('h1', null, 'Hello React !'));
14 }
15
16 export default Hello;
```

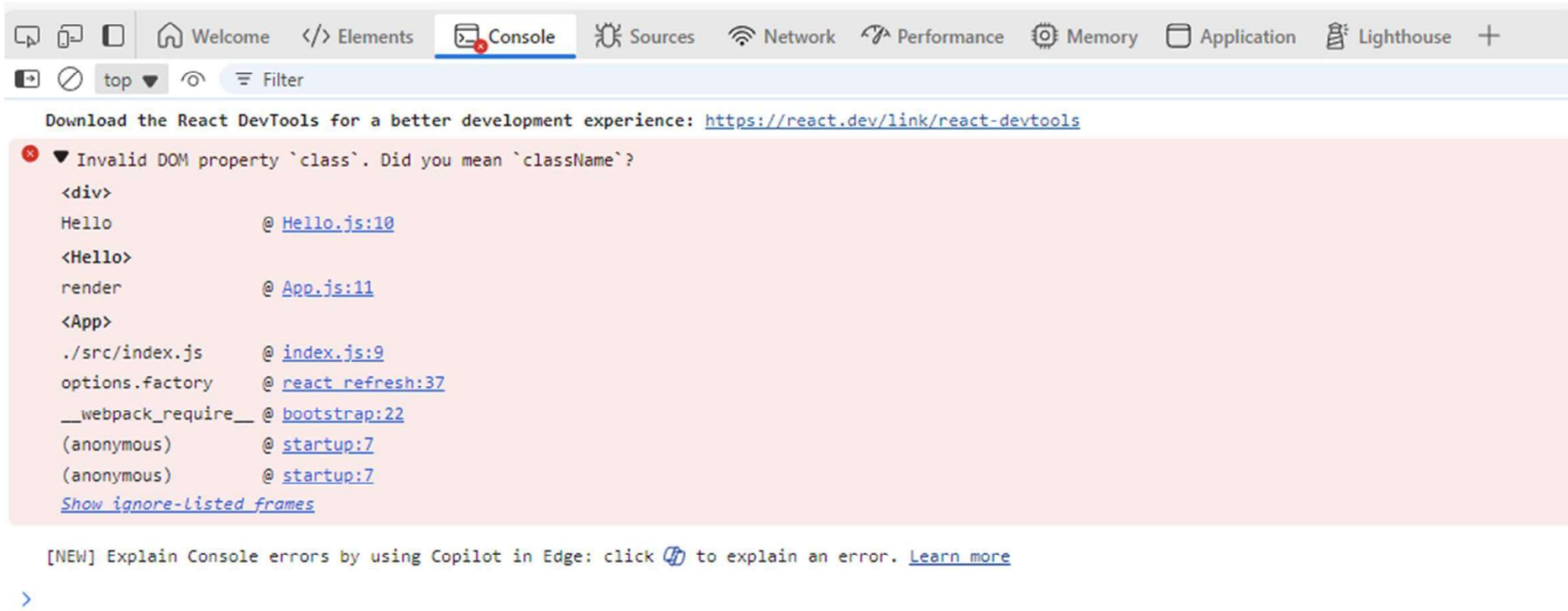
```
<!DOCTYPE html>
<html lang="en">
 <head> ... </head>
 <body>
   <noscript>You need to enable JavaScript to run this app.</noscript>
   <div id="root">
    <div class="App">
     <div id="helloDiv" class="dummy"> == $0
       <h1>Hello React !</h1>
     </div>
    </div>
   </div>
   <!--
       This HTML file is a template.
       If you open it directly in the browser, you will see an empty page.

       You can add webfonts, meta tags, or analytics to this file.
       The build step will place the bundled scripts into the <body> tag.

       To begin the development, run `npm start` or `yarn start`.
       To create a production bundle, use `npm run build` or `yarn build`.
     -->
 </body>
</html>
```

Error in Console

- In JS class is a Keyword, So Ended up with this.



JSX is Better Than Normal createElement()

- Our example just has two elements but imagine a component with ten or a hundred elements it starts to become really clumsy
- JSX on the other hand will keep the code elegant simple and readable

JSX Differences

- Follow the Link for More
- https://github.com/facebook/react/issues/13525

Class -> className

for -> htmlFor

camelCase property naming convention

- onclick -> onClick
- tabindex -> tabIndex

Summary

In this lesson, you should have learned how to:

- Creating Components
- Basic Component
- Nesting Components
- Props