

## Day4:React Component in details

15 December 2022 09:55 AM

### Different ways to create react components:

Create new file and give extension .js or .jsx both will work

When creating a React component, the component's name *MUST* start with an upper case letter  
e.g. HelloComponent.js or HelloComponent.jsx

#### 1)class syntax

```
import React from 'react'; //import React object from React package

//create class HelloComponent and HelloComponent should extends React.Component class
class HelloComponent extends React.Component {

}

export default HelloComponent; // export HelloComponentso that can be used in other component
```

If you run above code. You will get the

Error :No `render` method found on the returned component instance: you may have forgotten to define `render`.

It means every react class component requires a render method found ds

NOTE: Every render method should return something to output window

Render methos should return the JSX element to output window

Final code :	<pre>import React from 'react'; class HelloComponent extends React.Component {   render() {     // Return JSX Element     return &lt;h1&gt;Hello World!&lt;/h1&gt;;   } } export default HelloComponent;</pre>
--------------	--

import statement can be written in more better way:

Final code :	<pre>import React, { Component } from 'react'; class HelloComponent extends Component {   render() {     // Return JSX Element     return &lt;h1&gt;Hello World!&lt;/h1&gt;;   } } export default HelloComponent;</pre>
--------------	---

---

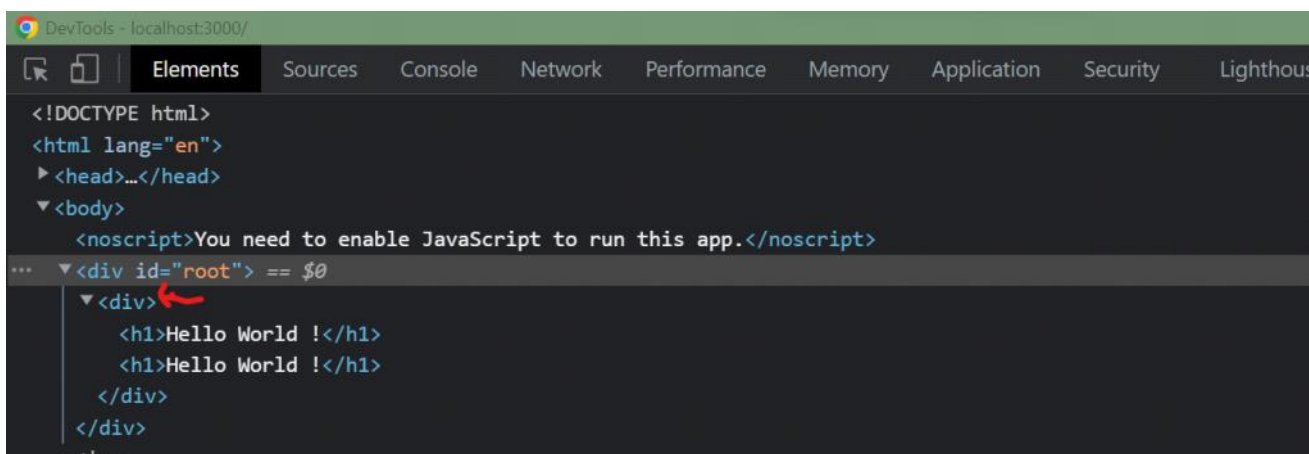
Render method can return only one jsx element at a time if you try to add more element in return statement you will get the error  
Error :No `JSX expressions must have one parent element.`

```
JS App.js 2, M HelloComponent.jsx 2, U X
src > components > day4 > HelloComponent.jsx > ...
1 import React from "react";
2 class HelloComponent extends React.Component {
3   render() {
4     return (
5       <h1>Hello World !</h1>
6       <h1>Hello World !</h1>
7     )
8   }
9 }
10 export default HelloComponent
```

so if you want return multiple element from render method wrap all element in one parent element like

```
import React, { Component } from 'react';
class HelloComponent extends Component {
  render() {
    // Should Return only one JSX Element
    return (
      <div>
        <h1>Hello World!</h1>
        <h1>Hello World Again!</h1>
      </div>
    );
  }
}
```

Above code will add the extra div in ui which does not required



To avoid this extra div element we have React **Fragment**

```
import React, { Component } from 'react';
class HelloComponent extends Component {
  render() {
    // Should Return only one JSX Element
    return (
      <React.Fragment>
        <h1>Hello World!</h1>
        <h1>Hello World Again!</h1>
      </React.Fragment>
    );
  }
}
```

This Fragment never render on ui

```

<html lang="en">
  <head>...</head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      <h1>Hello World !</h1>
      <h1>Hello World !</h1>
    </div> == $0
  <!--
    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
  -->

```

This Fragment can be create in different ways as well like

```

import React, { Component, Fragment } from 'react';
class HelloComponent extends Component {
  render() {
    // Should Return only one JSX Element
    return (
      <Fragment>
        <h1>Hello World!</h1>
        <h1>Hello World Again!</h1>
      </Fragment>
    );
  }
}

```

Or empty brackets

```

import React, { Component } from 'react';
class HelloComponent extends Component {
  render() {
    // Should Return only one JSX Element
    return (
      <>
        <h1>Hello World!</h1>
        <h1>Hello World Again!</h1>
      </>
    );
  }
}

```

-----Function syntax-----

Function declaration syntax	<pre> function HelloComponent() {   return (     &lt;&gt;       &lt;h1&gt;Hello World!&lt;/h1&gt;       &lt;h1&gt;Function Declaration Syntax!&lt;/h1&gt;     &lt;/&gt;   ); } export default HelloComponent; </pre>
Function Expression Syntax	<pre> const HelloComponent = function () {   return (     &lt;&gt;       &lt;h1&gt;Hello World!&lt;/h1&gt;       &lt;h1&gt;Function Expression Syntax!&lt;/h1&gt;     &lt;/&gt;   ); } export default HelloComponent; </pre>
Multiline Arrow Syntax	<pre> const HelloComponent = () =&gt; {   return (     &lt;&gt;       &lt;h1&gt;Hello World!&lt;/h1&gt;     &lt;/&gt;   ); } </pre>

	<pre>         &lt;h1&gt;Multiline Arrow Syntax!&lt;/h1&gt;       &lt;/&gt;     );   }   export default HelloComponent; </pre>
Singleline Arrow Syntax	<pre> const HelloComponent = () =&gt; (   &lt;&gt;     &lt;h1&gt;Hello World!&lt;/h1&gt;     &lt;h1&gt;Singleline Arrow Syntax!&lt;/h1&gt;   &lt;/&gt; ); export default HelloComponent; </pre>