ReactJS Components and JSX

Developing Applications using ReactJS

Objectives

- To understand what a ReactJS component is
- To understand what JSX is and why it is used
- To be able to create, export and import components using JSX

Components

- **ReactJS is fundamentally about Components**
  - Developers need to think in Components when working with ReactJS!
- **Components, according to the official ReactJS documentation:**

"React is a library for building composable user interfaces. It encourages the creation of reusable UI components which present data that changes over time."

- **Can be created using raw JavaScript**
  - Readability suffers!
- **React.Component is an abstract base class**
  - React components are typically subclasses that have a `render()` method (which is required)
- **Component names should always begin with a capital letter (by convention)**
  - Allows identification of native HTML tags and component tags
- **Facebook created JSX to help integrate HTML and JavaScript specifically for ReactJS**

25

Components – render()

- **Required in all components**
  - Can be representation of native DOM component (e.g. `<div />`)
  - Can be a custom composite component defined by the developer
  - Can return null or false to indicate that the component should not be rendered

- **Should be kept pure**
  - Does not modify component's state
  - Returns same result each time it is invoked
  - Does not directly interact with the browser

JSX

- **JavaScript Syntax Extension**
- **Not necessary to use but has some advantages:**
  - Makes it easier and quicker to create templates for developers familiar with HTML
  - Faster as optimisation occurs when compiling code to JavaScript
  - Type-safe with errors caught at compilation
- **Looks like normal HTML code in most cases**
- **Can use JavaScript expressions within it**

Project Set-up Changes

- **To use JSX files, project set-up needs to be slightly altered.**
- **Additional JSX file is needed**
- ***main.js* to be edited to use components created in JSX**
  - Any JSX files used need to be imported too

QAREACTJS

App.jsx file

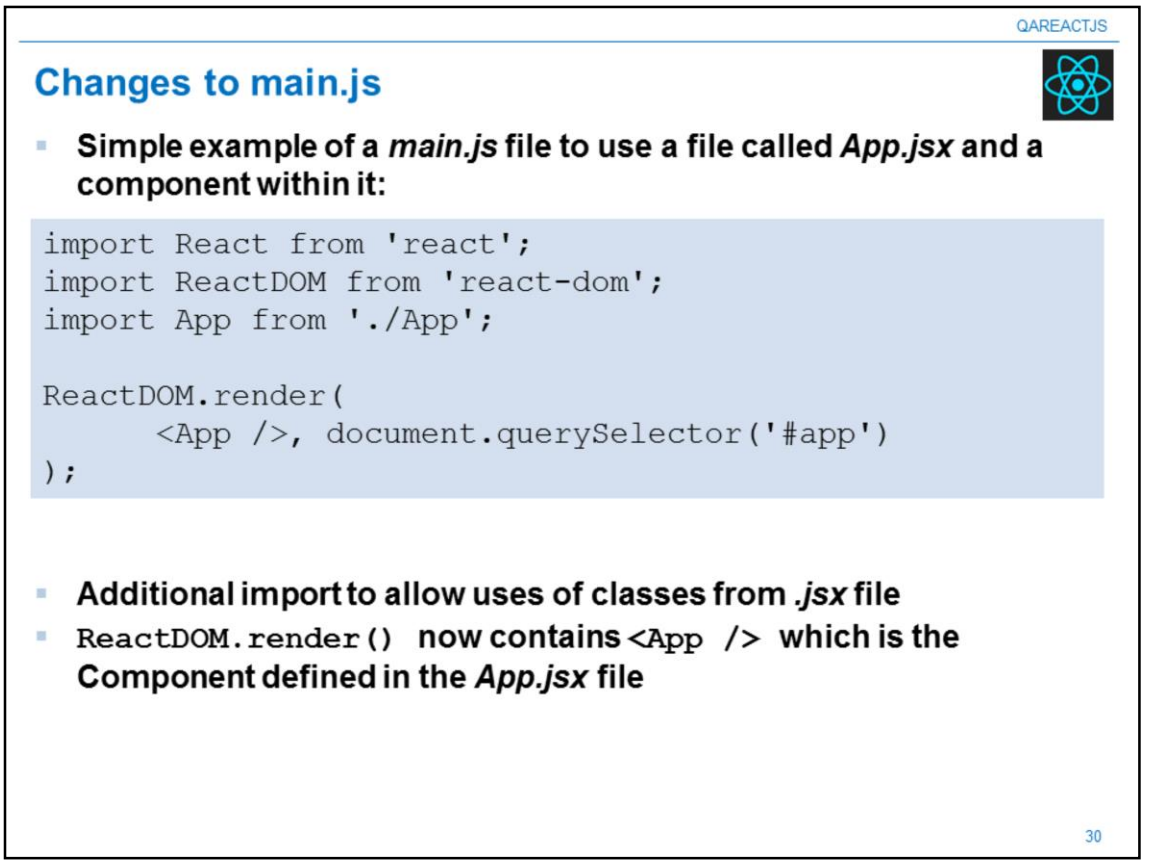
- **Simple example of an *App.jsx* and component within it:**

```
import React from 'react';

class App extends React.Component {
      render() {
          return (
            <div>
                  <h1>Hello, World!</h1>
                  <p>I am a paragraph</p>
            </div>
      );
}}

export default App;
```

- **ES2015 is used to import/export classes**
- **ES2015 class declarations and extensions used**
- **`render()` must have a wrapper element for multiple lines of HTML**

29

export default is used to export a single class from a JSX file.

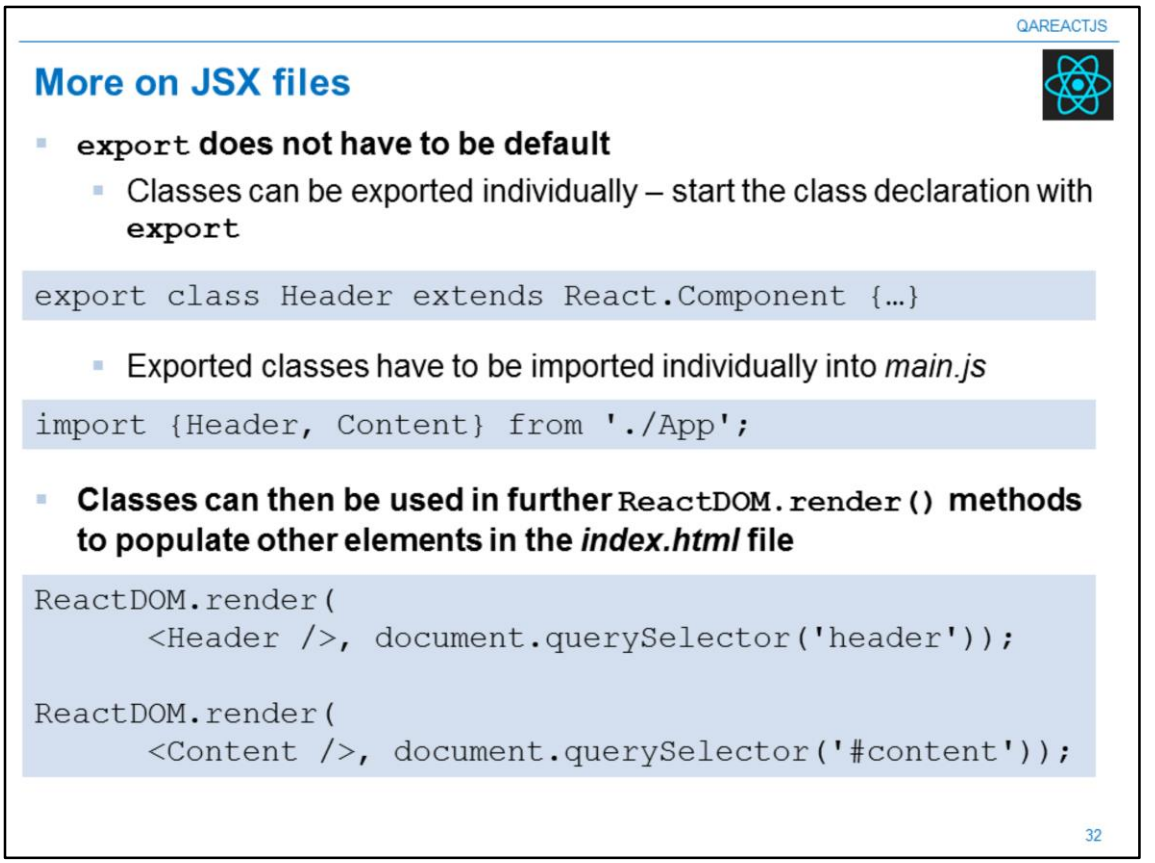


Multiple exports from a JSX require a slightly different syntax, which is covered later in this guide.

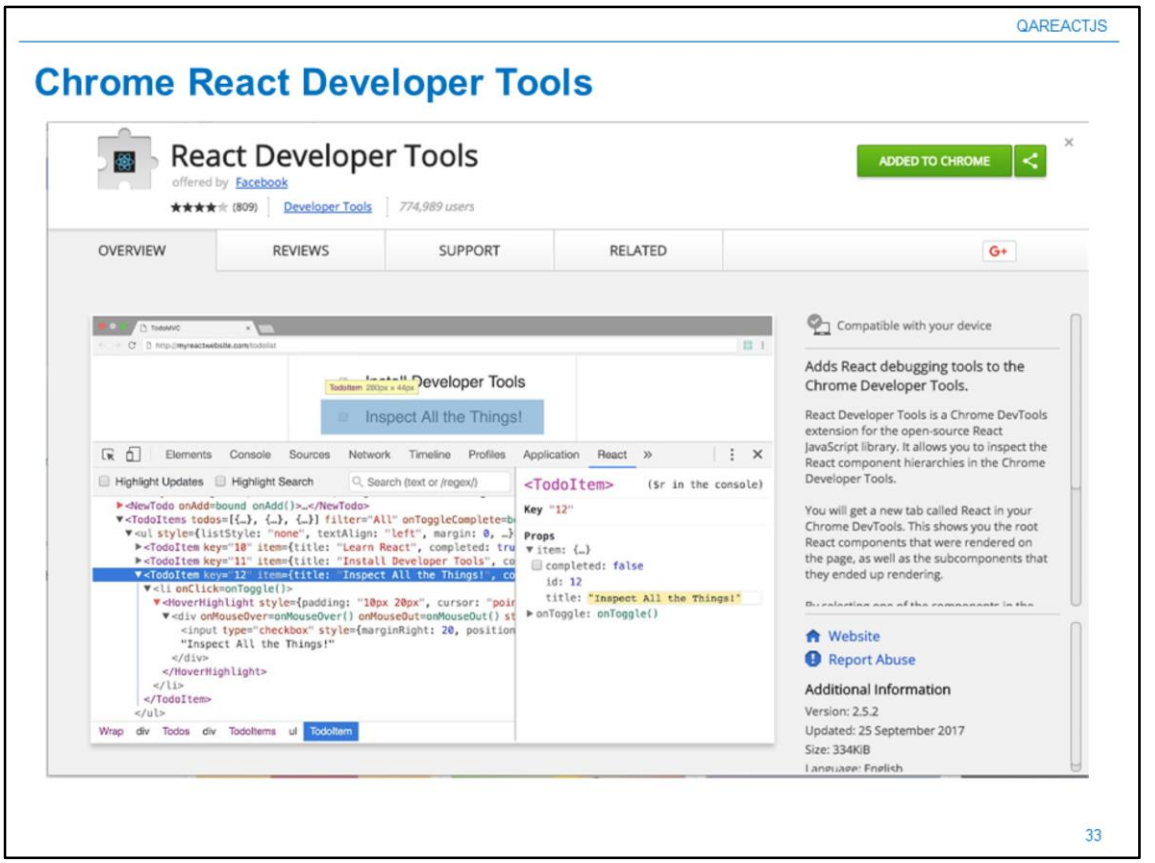
More on JSX files

- **Can contain more than one class**
  - Means that components can be split down into more manageable 'chunks'
- **Any classes defined can be declared and used within the exported component**
- **Example:**
  - *App.jsx* contains 3 classes: App, Header, Content
  - App class contains a Header component and a Content component

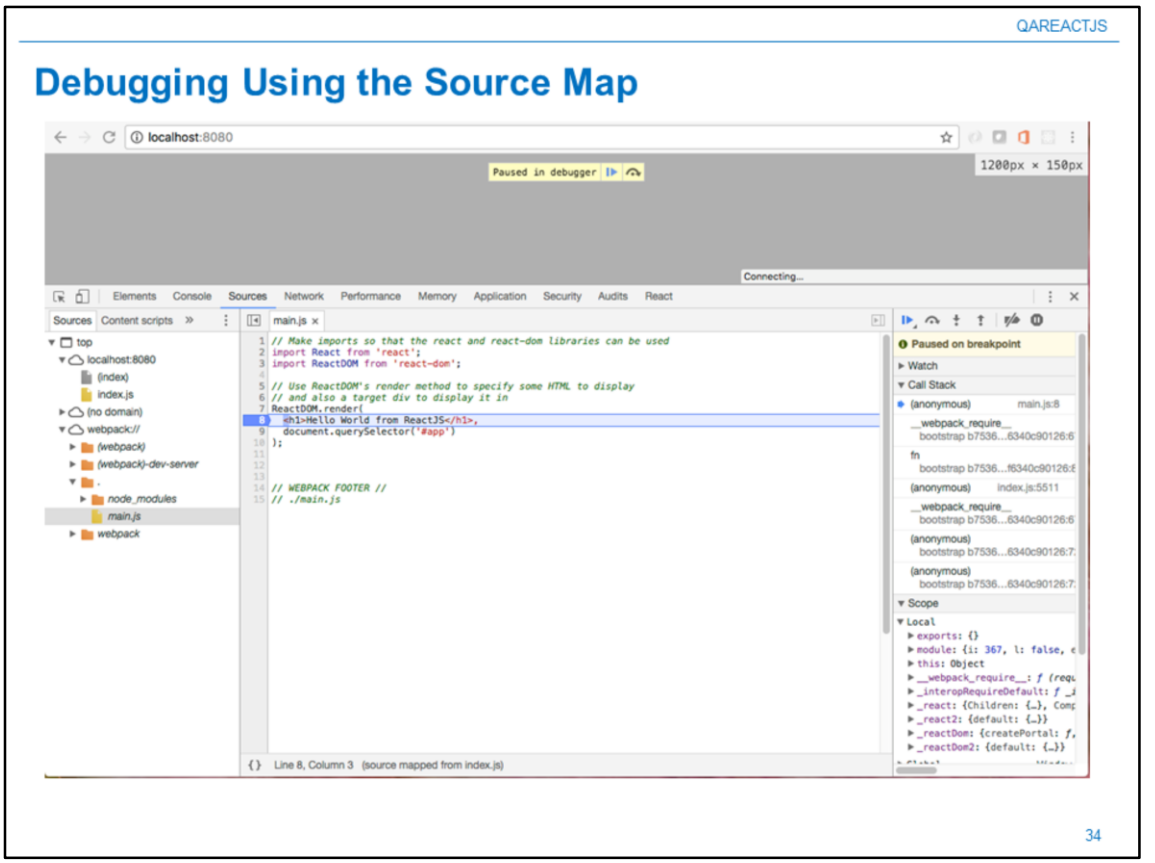
```
// Render method inside class App…
      render() {
          return (
            <div>
                  <Header />
                  <Content />
            </div>
          )
      }
```



Completing Exercise 3 and looking at the different solutions provided will lead to a thorough understanding of the use of multiple classes in a JSX file.



Allows developers to see information in the Developer Tools for React. It allows to access to how Components are rendered and also to their State, Props and other information, depending on its definition. (Image from https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi)



Using the devtool option of ‘source-map’ in the webpack.config.js file enables Chrome (and other browsers with advanced debuggers) to allow you to debug your JS and JSX files. They will appear in the . folder as shown on the image above. After selecting the appropriate file, you can set breakpoints in your React code and step and watch to your heart’s content.

Objectives

- To understand what a ReactJS component is
- To understand what JSX is and why it is used
- To be able to create, export and import components using JSX

Exercise Time

- **Complete EG03 – Creating Components with JSX**

Appendix – Defining Components pre-ES2015

- **Before ES2015 introduced classes to JavaScript, components were defined as follows:**

```
MyComponent = React.createClass ({
    render: function() {
      return <div>Some Content</div>
    }
}
```

- **This would produce a Component called MyComponent that could be rendered to the browser in the same way as using ES2015 classes:**

```
class MyComponent extends React.Component {
    render() {
      return <div>Some Content</div>;
    }
}
```