WEB COMPONENT DEVELOPMENT

Lect 1

- Welcome Back

- Gerard McCloskey
- Email gerard.mccloskey@atu.ie
- Room 3316

Course Overview

Description:

JavaScript frameworks facilitate website construction through the concept of component-based design. This module aims to leverage a component-based framework to assemble web applications.

Module Learning Outcomes:

On successful completion of this module the learner will be able to:

1. Elevate core language skills
2. Examine the RESTful architecture
3. Investigate a component-based website framework
4. Construct reusable website components
5. Build a significant data driven component-based website

Indicative Content

Indicative Content:

1. REST API

- Rationale for REST APIs
- REST API post requests
- Build a REST API server

2. React Component Framework

- ES6 primer
- Component classes & functions
- JSX & props
- React state & component lifecycle
- Event handling
- Conditionals
- Forms
- Router
- Pagination, Filtering, and Sorting
- Authentication and Authorization
- Deployment

RESOURCES

- *Additional libraries will be referenced in notes*
- *Blackboard / public folder*
  - *Slides*
  - *Code examples public folder*

- **Module Assessment:**
  - **100% CA**
  - **2 Assignments**
  - **1 Class test**

Resources:

Note: Learning resources may also be available on Blackboard.

Recommended Reading				
Author	**Year**	**Title**	**Publisher**	**ISBN**
Flanagan, D	2011	JavaScript: The Definitive Guide	O'Reilly	9780596805524
Freeman, E & Robson, E	2014	Head First JavaScript Programming	O'Reilly	9781449340131

Other Resources

Webography:

MDN JavaScript Documentation [https://developer.mozilla.org/en-US/docs/Web/JavaScript]
W3Schools JavaScript [http://www.w3schools.com/js/]
React [https://reactjs.org/, https://www.w3schools.com/react]
Node.js [https://nodejs.org/en/]

Getting Started

- Introduce React
- Review JavaScript with a detailed look at **objects** and **high order functions** in JavaScript.

```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((acc, n) => acc + n, 0);

console.log(sum);

function timesTwo(factor) {
  return function (x) { return x * factor; };
}

const double = timesTwo(2);
console.log(double(5));
```

What is React

- React is a JavaScript library created and maintained by Facebook
- UI library / Framework
- Build websites in an organised way with reusable components.
- Uses something called JSX which we will look at next week

- Previously we had our HTML, CSS and JavaScript logic is separate files (separation of concerns).
- React component approach see the HTML, CSS and JavaScript logic back together in our component.

Prerequisites

- Basics of JavaScript
  - Data structures, arrays, objects, loops, functions, etc.
  - Document Object Model(DOM)
- Builtin JavaScript functions like `forEach`, `map`, `filter`, `find` etc
- Arrow functions
- Asynchronous Programming use of promises.
- async/await
- NPM or Node Package Manager

Why use React

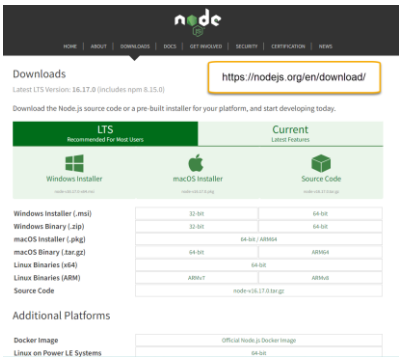
- Components / organisation
- Flexibility
- Support
- Performance uses something called the virtual DOM

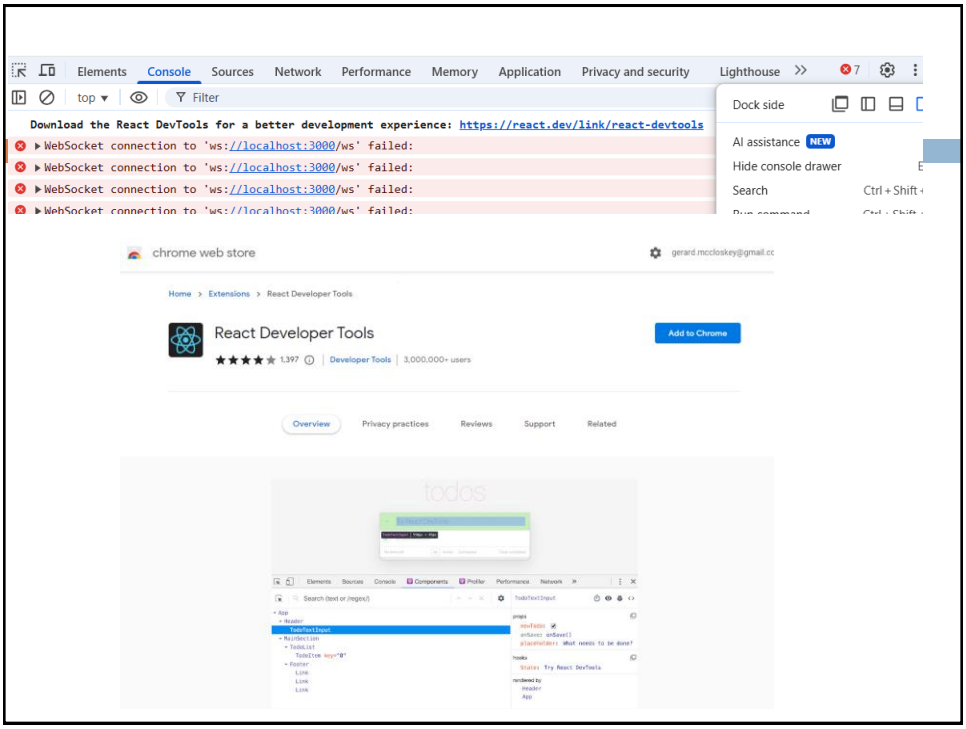
- React has four key element
  - Components
  - Props
  - State
  - Events

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Hello World Example

- Need
- Node.js
- NPM
- IDE :Visual code editor





Quick start node package called *Create React App*.

- Make/open a new folder in Visual Studio Code

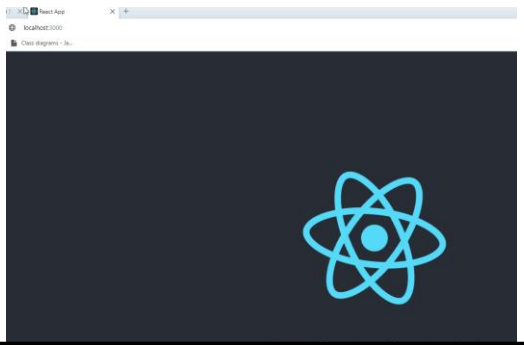
```
C:\>mkdir demo1_react

C:\>cd demo*

C:\demo1_react>npx create-react-app my-app
```

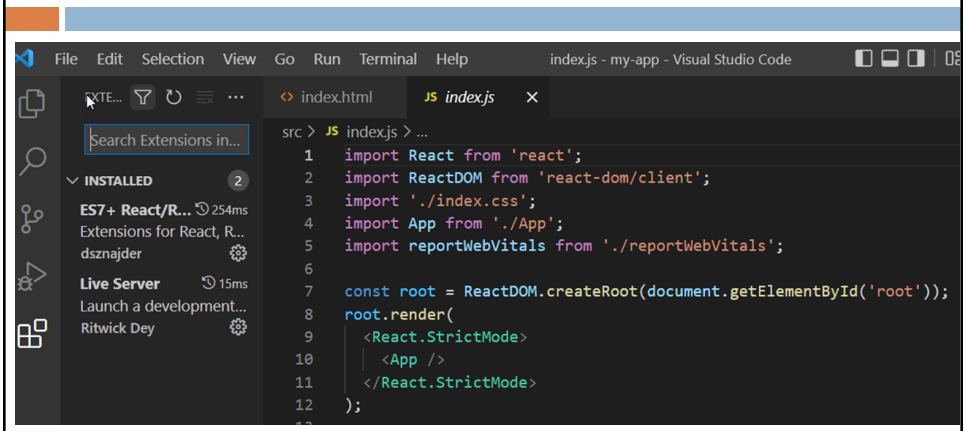
Can't use uppercase letters.

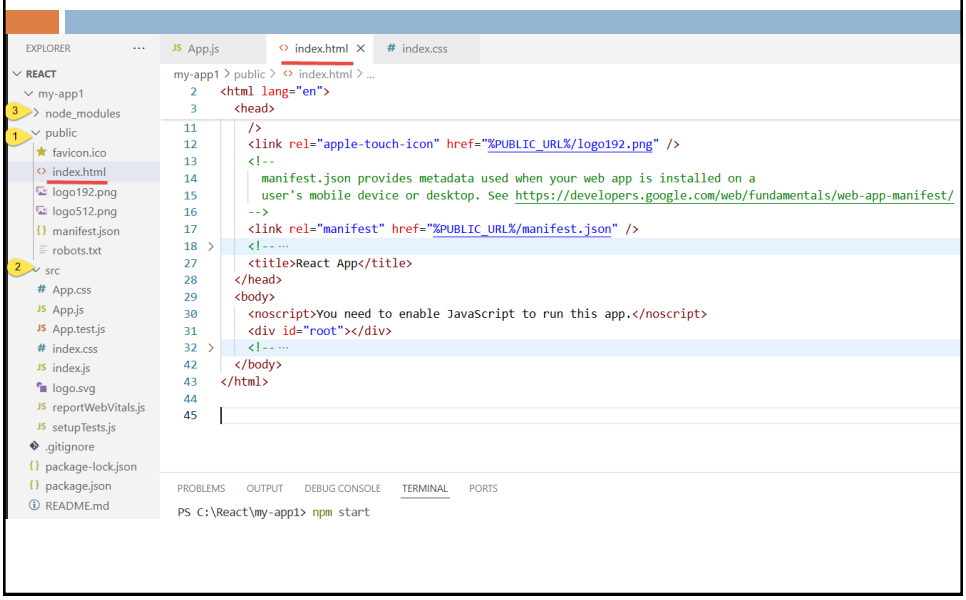
```
C:\demo1_react>npx create-react-app my-app

Creating a new React app in C:\demo1_react\my-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[                    ] / idealTree:my-app: sill idealTree buildDeps
```

```
Success! Created my-app at C:\demo1_react\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!
npm notice
npm notice New minor version of npm available! 8.12.1 -> 8.19.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.19.2
npm notice Run npm install -g npm@8.19.2 to update!
npm notice
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! htt

PS C:\demo1_react\my-app> npm start
```




File Edit Selection View Go Run Terminal Help
index.js - my-app - Visual Studio Code
EXTE...
Search Extensions in...
INSTALLED 2
ES7+ React/R... 254ms
Extensions for React, R...
dsznajder
Live Server 15ms
Launch a development...
Ritwick Dey
index.html
index.js
src > index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9 <React.StrictMode>
10 <App />
11 </React.StrictMode>
12);


EXPLORER
REACT
my-app1
node_modules
public
favicon.ico
index.html
logo192.png
logo512.png
manifest.json
robots.txt
src
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md
App.js
index.html
index.css
my-app1 > public > index.html > ...
<html lang="en">
<head>
/>
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<!--
manifest.json provides metadata used when your web app is installed on a
user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!-- ···
<title>React App</title>
</head>
<body>
<noscript>You need to enable JavaScript to run this app.</noscript>

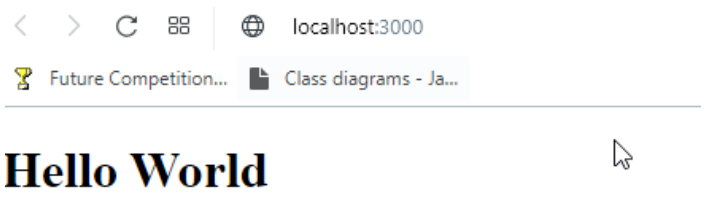
<!-- ···
</body>
</html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\React\my-app1> npm start

- `index.html` --> `<div id="root"></div>`.
- `index.js` calls `ReactDOM.render(..., document.getElementById('root'))`.
- Webpack bundles `index.js` (plus imports) into a single JavaScript file.
- That bundle is automatically injected into index.html during build

- React applications are made up of a hierarchy of components.
- `App` is the top-level component.
- Only default application generated by Create React App

src > JS index.js

```
1  import React from 'react'
2  import ReactDOM  from 'react-dom'
3
4  ReactDOM.render(<h1>Hello World</h1>, document.getElementById('root'))
5
```



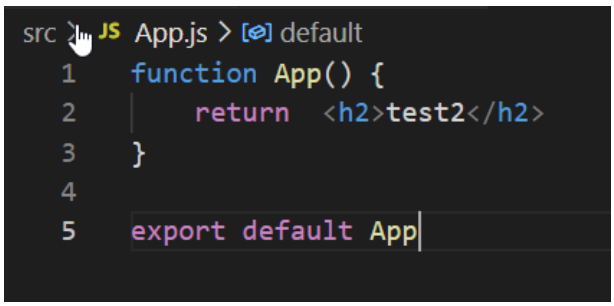
- Now change the page to display

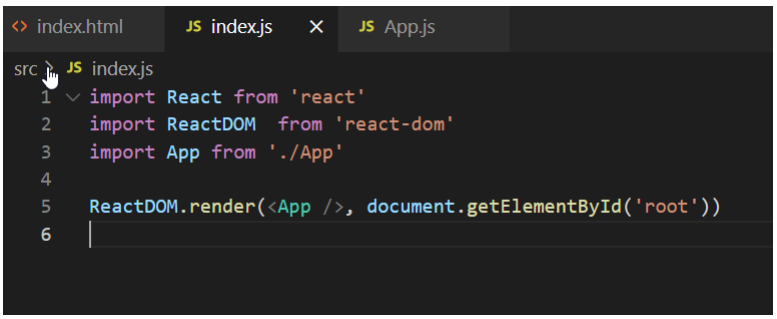
Hello World

welcome back

src > JS index.js

```
1  import React from 'react'
2  import ReactDOM  from 'react-dom'
3
4  ReactDOM.render(<h1>Hello World</h1>, document.getElementById('root'))
5
```


src > App.js > default
1 function App() {
2 return <h2>test2</h2>
3 }
4
5 export default App


index.html
index.js
App.js
src > index.js
1 import React from 'react'
2 import ReactDOM from 'react-dom'
3 import App from './App'
4
5 ReactDOM.render(<App />, document.getElementById('root'))
6

JSX

- JSX: XML/HTML markup syntax that's embedded into JavaScript code.
- JSX is the language used to describe UIs.
- In react this allows us to create and reuse custom components (core feature).
- JSX is rendered to the DOM node supplied to ReactDOM via the **render()** function, which accepts it as an input.createRoot()
- JSX is not directly understood by web browsers and needs to be transformed into standard JavaScript

Transpiling

- Converts JS programming code from one version into another version.
- Not all web browsers support the same set of new JS features.
- Using a JavaScript transpiler like Babel, the latest version of JS code can be transpiled to run on any web browser.

Tutorial recap excercises

- Modules
- Default parameters
- Rest parameters ...
- Spread operator ...
- Destructuring assignment
- Template string literals
- Set, Map, async programming

- The **rest parameter** syntax allows a function to accept an indefinite number of arguments as an array

```
function sum(...theArgs) {
  let total = 0;
  for (const arg of theArgs) {
    total += arg;
  }
  return total;
}

console.log(sum(1, 2, 3));

console.log(sum(1, 2, 3, 7));
```

- The spread (...) syntax allows an iterable, such as an array or string, to be expanded in places where zero or more arguments (for function calls) or elements (for array literals) are expected.

```
const numbers = [1, 2, 3, 4, 5];
console.log(sum(...numbers));
```

Summary

- Outline course and CA what's expected this term.
- Need to review JS with emphasis on high order functions.
- Created a small React application to understand key components.