Before we begin...

- Open up these slides:
  - https://bit.ly/2roZ9Na

Classes & React

Learning Objectives

- **Understand** the class syntax in JavaScript
- **Effectively** use classes in JavaScript
- **Understand** Webpack and its role in web development
- **Effectively** use Webpack
- **Understand** React and its history
- **Identify** and **understand** Components in React

Agenda

- Classes
- Webpack
- React

A quick review

- Modules
- Transpilation & Compilation
- Babel
- Webpack

Projects Time!

Classes

GENERAL ASSEMBLY

What are classes?

- Syntactic sugar for inheritance in JavaScript
- An ES2015 feature!
- Modular JavaScript is made a little easier with these features
- There are two main things:
    - The class itself (the *blueprint*)
    - The instances (the actual houses built from it)
- You can extend classes for complex inheritance too!

How to create classes?

```
class Person {
    constructor() {
        console.log("A person was just born!
    }

    sayHiTo(name) {
        console.log(`Hello ${name}`);
    }
}
```

@threequal

How to create classes?

```
class Person {
    constructor(name) {
        this.name = name;
        console.log(`${this.name} was just born!`);
        console.log(this);
    }

    sayHiTo(name) {
        console.log(`Hello ${name}, I'm ${this.name}
    }
}
```

@threequal

How to create instances?

```
class Person {
    constructor(name) {
        this.name = name;
        console.log(`${this.name} was just born!`);
    }
    sayHiTo(name) {
        console.log(`Hello ${name}, I'm ${this.name
    }
}

const jane = new Person("Jane");

jane.sayHiTo("Bill");
```

@threequal

Inheritance and Classes

```
class Shape {
    constructor(type) {
        this.type = type;
        console.log(`A ${type} was createc
    }
}

class Rectangle extends Shape {
    constructor(width, height) {
        super("Rectangle");
        this.width = width;
        this.height = height;
    }
    getArea() {
        return this.width * this.height;
    }
}
```

@threequal

Resources

- [MDN: Classes]
- [Exploring JS: Classes]
- [Codecademy: Classes]

Webpack

GENERAL ASSEMBLY

What is Webpack?

- It is a *Build System* and a *bundler*
- It automates tasks for us
- It takes our code, transforms and bundles it, then returns a new version of our code
- We need to make sure our code is browser compatible:
    - SCSS -> CSS
    - ES2015 -> JavaScript

What is Webpack?

- It doesn't do anything by default
- But can be extended to do lots of other things:
  - Minifying and Optimizing Code
  - Minifying Images
  - etc.
- Before this, we have to add lots of scripts if our code is broken up - Webpack brings all of our code together

Why do you need it?

- It helps structure our code
- It organises and automates the tasks we need to do
  - e.g. using Babel
- It saves us from having to combine files ourselves
- It helps us work with larger applications (e.g. by splitting code)
- It can help create our server, can replicate different environments (e.g. development or production) and can add **Hot Module Replacement**

Any alternatives?

- Parcel
- FuseBox
- RollUp
- Browserify
- Grunt
- Gulp
- Make
- Using NPM as a Task Runner

What it needs to know

- The starting point of your application
- What transformations it needs to perform
- The "mode" (whether it is development or production)
- Where it should save your transformed code

We define all of this in a file called *webpack.config.js*

Some Webpack concepts

- **entry** - Where your application starts
- **output** - Where your resulting code goes
- **loaders** - A single transformation/process (e.g. Babel)
- **rule** - All transformations that need to take place for certain files
- **bundle** - Your transformed code (once it is combined)
- **mode** - The current environment (development or production)

webpack.config.js example

```
const config = {
  entry: ["./app/js/index.js"],
  output: {
    path: __dirname + "/dist",
    publicPath: "/",
    filename: "bundle.js"
  },
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        exclude: /node_modules/,
        loaders: ["babel-loader"]
      }
    ]
  }
};

module.exports = config;
```

@threequal

ES2015 to ES5

Resources

- TinselCity: whys:packers
- Webpack
- SurviveJS
- Webpack Academy

React

GENERAL ASSEMBLY

What is React?

- A JavaScript library for building user interfaces
- Include it before your own code:
    - Script
    - NPM
- Built by Facebook
- It is a Front-end JavaScript Framework
    - It changes the way we write code
    - Other frameworks: Vue, Angular, Backbone etc.

The React Ecosystem

- You can go straight ahead and write React code, though it is often written with other technologies:
  - Webpack
  - Babel
  - React (and ReactDOM)
  - JSX (JSX in Depth)
  - ...

What is JSX?

- A syntax extension to JavaScript (not a part of ECMAScript or anything though)
- It's almost like HTML in JavaScript
- JSX produces React "elements"
- React doesn't require JSX, but it is suggested
- You can embed expressions in JSX (like interpolation)
- It just makes our lives easier!

What is JSX?

```
<h1 id="hello">Hello World</h

// Compiles to...

React.createElement(
    "h1",
    { id: "hello" },
    "Hello World"
);
```

@threequal

What is JSX?

```
<img src="http://fillmurray.com/400/400" id="bill">

// Compiles to...

React.createElement(
     "img",
     { src: "http://fillmurray.com/400/400", id: "bill
     null
);
```

@threequal

What does React teach?

- Declarative
- Unidirectional Data Flow
- Composition
- Explicit Mutations
- Remember that it is just JavaScript

Imperative vs. Declarative

- Imperative is telling a computer how to do something
- Declarative is telling a computer what to do
    - It relies on the magic
    - Declarative is all about abstraction
    - React Components are always declarative

Advantages

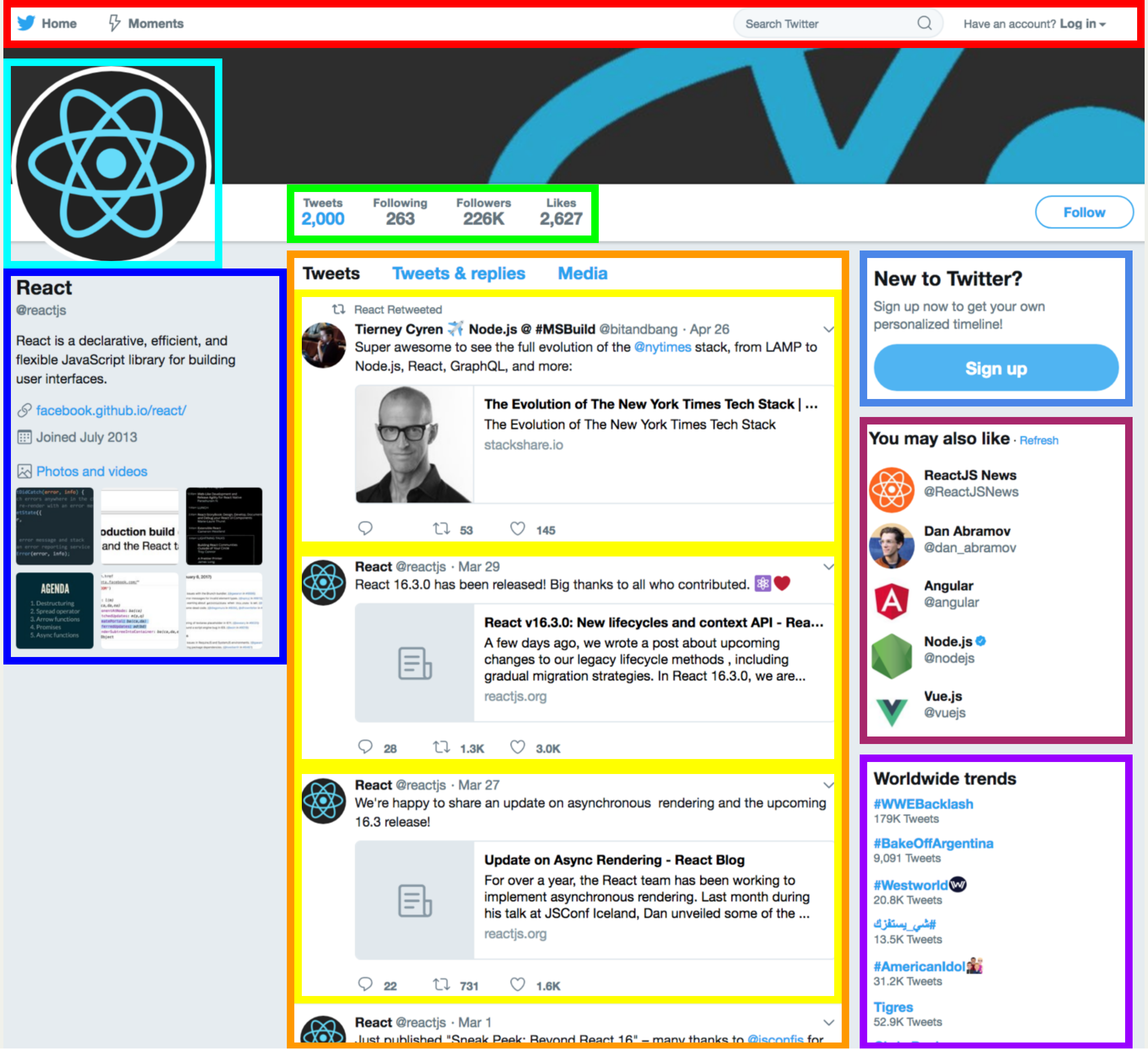
- Really easy to see the structure of your app
- Very good at managing state
- Performant
- Virtual DOM
- Data Binding
- Easy to test
- Isomorphic (can be rendered server-side)
- Agnostic (you can use it with all sorts of other libraries as React is just the view layer)
- Learn once, write everywhere

Disadvantages

- A big library
- Lots of magic
- It is just the view layer
- Typically requires a transformation step
- A steep learning curve
- It changes incredibly regularly

Working with Components

- A component is one of the fundamental parts of React
- Each component represents a small part of a page
  - And each component manages their own state
- You compose your app with lots of different components



Tweets 2,000 · Following 263 · Followers 226K · Likes 2,627

React
@reactjs

React is a declarative, efficient, and flexible JavaScript library for building user interfaces.

facebook.github.io/react/

Joined July 2013

Photos and videos

Tweets | Tweets & replies | Media

React Retweeted
Tierney Cyren ✈️ Node.js @ #MSBuild @bitandbang · Apr 26
Super awesome to see the full evolution of the @nytimes stack, from LAMP to Node.js, React, GraphQL, and more:

The Evolution of The New York Times Tech Stack | ...
The Evolution of The New York Times Tech Stack
stackshare.io

53 · 145

React @reactjs · Mar 29
React 16.3.0 has been released! Big thanks to all who contributed. ⚛️❤️

React v16.3.0: New lifecycles and context API - Rea...
A few days ago, we wrote a post about upcoming changes to our legacy lifecycle methods , including gradual migration strategies. In React 16.3.0, we are...
reactjs.org

28 · 1.3K · 3.0K

React @reactjs · Mar 27
We're happy to share an update on asynchronous rendering and the upcoming 16.3 release!

Update on Async Rendering - React Blog
For over a year, the React team has been working to implement asynchronous rendering. Last month during his talk at JSConf Iceland, Dan unveiled some of the ...
reactjs.org

22 · 731 · 1.6K

React @reactjs · Mar 1
Just published "Sneak Peek: Beyond React 16" – many thanks to @jsconfis for

New to Twitter?
Sign up now to get your own personalized timeline!
Sign up

You may also like · Refresh

ReactJS News	@ReactJSNews
Dan Abramov	@dan_abramov
Angular	@angular
Node.js	@nodejs
Vue.js	@vuejs

Worldwide trends

#WWEBacklash	179K Tweets
#BakeOffArgentina	9,091 Tweets
#Westworld	20.8K Tweets
#شي_يستفزك	13.5K Tweets
#AmericanIdol	31.2K Tweets
Tigres	52.9K Tweets

Let's get into it!

Our Hello World!

```
const React = require("react");
const ReactDOM = require("react-dom

ReactDOM.render(
    <h1>Hello World</h1>,
    document.querySelector("#app")
);
```

Our First Component

```
const React = require("react");
const ReactDOM = require("react-dom");

class HelloWorld extends React.Component
  render() {
    return <h1>Hello World</h1>;
  }
}

ReactDOM.render(
    <HelloWorld />,
    document.getElementById("root")
);
```

@threequal

Interpolation with JSX

```
const React = require("react");
const ReactDOM = require("react-dom");

class FavNumber extends React.Component {
  render() {
    const favNumber = 42;
    return <h1>Favourite Number: {favNumber}!</h1>;
  }
}

ReactDOM.render(<FavNumber />, document.getElementById("roo
```

@threequal

Passing Data as Props

```
const React = require("react");
const ReactDOM = require("react-dom");

class Hello extends React.Component {
  render() {
    return <h1>Hello {this.props.name}!</h
  }
}

ReactDOM.render(
    <Hello name="Bill" />,
    document.getElementById("root")
);
```

@threequal

Component Composition

```
const React = require("react");
const ReactDOM = require("react-dom");

class TodoItem extends React.Component {
  render() {
    return <li>{this.props.task}</li>;
  }
}

class TodoList extends React.Component {
  render() {
    return (
      <ul>
        <TodoItem task="Task One" />
        <TodoItem task="Task Two" />
        <TodoItem task="Task Three" />
      </ul>
    );
  }
}

ReactDOM.render(<TodoList />, document.getElementById("roo
```

@threequal

Events

```
const React = require("react");
const ReactDOM = require("react-dom");

class Hello extends React.Component {
  render() {
    const { name } = this.props;
    return (
        <h1 onClick={() => alert(`${name} was clicked!`)}>
            Hello {name}!
        </h1>
    )
  }
}

ReactDOM.render(<Hello name="Bill" />, document.getElementById("roo
```

@threequal

Resources

- ReactJS Website
- Egghead: Beginner's Guide to React
- React for Beginners
- Codecademy
- Cabin: Learn React
- React Armory: Learn React
- The Road to Learn React
- SurviveJS: React

Homework

- It's Project Time!
- Do some React Tutorials
- Add Babel and Webpack to previous homework!
- Read up on ES2015
-
  - Translate some of your previous code into it!
- Finish all exercises from class
- Upload your homework to GitHub
- Prepare for next lesson

Homework (Extra)

- Go through some tasks in Exercism
- Get into JavaScript30
- Go through The Modern JavaScript Tutorial
- Read Exploring ES6
- Read Eloquent JavaScript
- Read Speaking JavaScript

What's next?

- More React
  - Events
  - State
  - Lifecycle Methods

Questions?

Feedback time!

Lesson 14: ***Classes and React***

https://ga.co/js05syd

Thanks!