React

Away we go!

By Stephen Lampa & Jesse Williamson

Announcement!
(sad news)

Moving next week's session to Wednesday

(because I want to see you guys sooner obvs)

Any lingering JavaScript questions?

Today’s Topics

- What is Front-End Development?
- What is React?
- create-react-app
- npm
- webpack
- Babel
- Installfest?

What is Front-End Development?

Creating web applications that **users** can view and interact with!

What is React?

React: A view library created by Facebook (yay!)

- Doesn’t have strong opinions
- Is there for display purposes

Brief History

- Component design leads to easier/better composition
- Virtual DOM makes real time applications faster
- Some legal troubles early on (3-clause BSD vs MIT)

Virtual DOM

- HTML pages used to have to reload the entire page
- A copy of the DOM is compared against
- Only re-renders the changes

Components

- Come back to this

create-react-app

The easiest way to get up and running

What is create-react-app?

The quickest and most up-to-date way of standing up a React application

Create a React application

```
$ npx create-react-app [AppName]
```

Some notes:

- Please make sure to have node and npm installed!
- More specifically npm must be of version 5.2+
    - To find out the version, please run `$ node -v` in PowerShell or terminal

Run a React application

```
$ cd [AppName]

$ npm run start
```

Some notes:

- This should automatically open up your browser to \`localhost:3000\`
- If not, open a browser and type in \`localhost:3000\` in the address bar

/public directory

- The most important file is index.html
  - This is the HTML that will be served in the browser
  - Holds all the scripts/link tags
- A place for special files not found in builds (libraries incompatible with Webpack)
- A place for assets (like images, videos, etc)

/src directory

- The most important file is index.js
  - This is the entry point for your React code
  - This is what is injected into public/index.html
- This is where all your code will live! (JS, JSX, CSS, etc)

npm

Where [noun + ‘.js’] is one line away!

npm

The way to manage your life (and JavaScript)

Node Package Manager (npm)

- Defined as a package manager first, task runner second
  - Package Manager: Manages all dependencies
  - Task Runner: Can run scripts

package.json

Includes:

- The metadata of a project
- Dependencies (for things needed at runtime)
  - devDependencies (for things that don't need to be shipped (ie testing, linting, etc)
- Scripts

Basic npm commands

```
$ npm run <command>
```

- start: Starts a local version of your app
- build: Builds a production ready version of your app
- test: Tests given your test runner

npm commands

```
$ npm run <command>
```

- Can also run whatever kinds of scripts you decide to include

npm commands

```
$ npm install [package]
```

- How to get the dependencies necessary for your project
- Saves version of dependency in your package.json (if missing)
- Run `--save-dev` for devDependency

npm commands

```
$ npm help
```

- Shows all possible commands you can run

node_modules

- Where all your dependency libraries are downloaded to
- Can traverse to see source code
  - This is sometimes necessary
- No need to check this into version control

If you want to feel pain…

Or you’re just curious

npm commands

```
$ npm run eject
```

- Only do this if you know for sure you want to see/modify the config files and dependencies. THIS IS A PERMANENT ACTION.

Babel

Or how you deal with Safari and Internet Explorer

babel

- A tool that compiles/transpiles ES6+ (or ECMAScript 2015+) code to be compatible with very old browsers
- Generally allows bleeding edge versions of ECMAScript to be run on any browser

Webpack

It’s pretty neat

Brief Overview

- In a few words, it takes all your files and makes them into smaller files
- Makes use of a dependency graph and builds/bundles based on set of rules

Loaders

Modules that help load/build files at an individual level to eventually bundle based on the loader rules. Can be used on pretty much any file type (just need a loader library for it)

Example:

- If you have files with ES6 + Typescript it can convert it into a single ES5 file

Plugins

Like loaders, but usually are used at the end of a bundle. This includes things like minification or file code optimization

Example:

- If you have files with ES6 + Typescript it can convert it into a single ES5 file

Questions?