Before we begin...

- Videos On!

Welcome

Agenda

- Parcel
- Modules

Parcel

What is Parcel?

Parcel is a zero-configuration command line tool, that provides a lot of functionality to us. It is a:

- Build system
- Bundler
- Transpiler / Compiler
- Plus more

Build System

Parcel will automate tasks for us, such as:

- Preparing our apps for production
- Starting and stopping servers
- Testing and linting our code

Bundler

Parcel will combine multiple files into one for us (in a very intelligent way)

Transpiler / Compiler

Parcel will take our code, perform some transformation for us and then return a new version of it (often using a tool called Babel). The main purpose of this is for **browser compatibility**.

Examples transformations include:

- SCSS to CSS
- New versions of JS to old versions of JS
- Fancy JS (such as React) to regular JS
- Jade to HTML
- Large images to optimized images (often using new formats)

More

- Hot Module Reloading
- Code Splitting
- Tree Shaking
- etc.

Alternatives to Parcel

- Webpack
- ESBuild
- Vite
- Grunt
- Gulp
- Snowpack
- FuseBox
- RollUp
- Browserify
- Plus, many more

Using Parcel

Installing Parcel

```
# Create a folder called `my-js-app`
mkdir my-js-app
# Move into the `my-js-app` folder
cd my-js-app
# Set up the new NPM project
npm init
# Add Parcel as a development dependency
npm install --save-dev parcel
```

Starting our Build System

Add a start script in your `package.json`

```
{
  "name": "parcel-install",
  "version": "0.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "parcel app/index.html"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Now run `npm run dev` in your terminal

Building for Production

Add a script in your `package.json`

```
{
  "name": "parcel-install",
  "version": "0.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "parcel app/index.html",
    "build": "parcel build app/index.html"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Now run `npm run build` in your terminal

Modules

What are Modules?

Modules allow us to build big JavaScript projects by allowing us to easily work in multiple files, and to bring in any dependencies that are necessary:

- We can export code from one file
- And we can require (essentially import) any files or dependencies that we need in another file

They only work with certain tools (e.g. with Parcel, or Node etc.)

Exporting Code

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

export default speak;
```

Exporting Code

```
function add(x, y) {
  return x + y;
}

function subtract(x, y) {
  return x - y;
}

export {
  add: add,
  subtract: subtract
}
```

Importing Code

```
// Import the default export from hello.js
//   Call it `sayHello`
import sayHello from "./hello";

// Import add and subtract exports from maths.js
import { add, subtract } from "./maths";
```

That's all for tonight!

Homework

- Read JavaScript.info's import and export page
- Read Thinking in React
- Read Introducing JSX
- Go through the React Docs Quick Start (use Parcel for this)

What's Next?

- JSX
- React
  - Components
  - Hooks

Thank You!