Handout

Diving into ES2015

Lesson 1 - Step 2

Darin Haener

Lesson 1.2 - Tools of the Trade

Text Editor

You're going to need a basic text editor for this course to follow along with some of the code samples. These days I prefer Vim, but there are a lot of great choices out there:

- Atom - a great free choice built by GitHub
- Sublime Text - another great choice, similar to Atom, but not 100% free
- Webstorm - a full blown JavaScript IDE, but it comes at a price.

The most important thing here is that you use a text editor that you are comfortable with.

Node and npm

The next thing we're going to need to have installed is Node and npm. Luckily, npm comes bundled with Node when installed from the Node website. If you don't already have it, let's go ahead and install the latest version of Node right now.

Everything you need is available on the NodeJS website. If you're using Mac or Windows, you're in luck, as there are installers for both platforms. Just make sure you click on the "Stable" button before downloading to ensure that you're getting the latest stable version.

If you're using Linux, you can download the Linux binaries. Installing from the binaries is fairly straightforward:

```
$ wget https://nodejs.org/dist/v5.7.1/node-v5.7.1-linux-x64.tar.xz
$ tar xzf node-v5.7.1-linux-x64.tar.xz
$ sudo cp -rp node-v5.7.1-linux-x64 /usr/local/
$ sudo ln -s /usr/local/node-v5.7.1-linux-x64 /usr/local/node
```

Babel

In order to take advantage of all of the new ES2015 features we're going to use the Babel CLI to run our Node REPL, and to run our scripts.

What is Babel? Simply put, Babel is a JavaScript compiler. Babel takes code that is written in future versions of JavaScript (or things like JSX for React) and compiles it to code that is compatible in most modern browsers.

You may be wondering what the heck a Node REPL is right now. REPL stands for Read-Eval-Print-Loop. It is simply an interactive environment for writing and testing code for a computer language.

Let's go ahead and install the Babel CLI now that we have Node and npm installed:

```
$ npm install -g babel-cli
```

We're using the `-g` flag to tell npm to install this package globally so that we will have access to the CLI commands everywhere.

Normally we would use the node command to enter into a Node REPL, but because we want to take advantage of all of the newest ES2015 features, we're going to be using the `babel-node` command. This was installed when we installed the Babel CLI. If you're able to type in `babel-node` without any errors you're ready to move on and start writing some ES2015 code!

In order to use all of the ES2015 features available to us, we're going to need to install the ES2015 presets module. Let's go ahead and create a directory where we can store all of our code samples from this course. After creating the directory, we'll install the babel presets module:

```
$ mkdir es2015
$ cd es2015
$ npm init -y
$ npm install babel-preset-es2015
```

We could include these presets every time we start the REPL like this:

```
$ babel-node --presets es2015
```

But it's much easier to create a `.babelrc` file to handle this for us! A `.babelrc` is simply a configuration file that Babel will look for any time it is run. Let's go ahead and do that in the root of your code directory that we created.

```
{
  presets: [
    'es2015'
  ]
}
```