



Now that we have the main software installed, it's time to install the plugins that we'll be using with Gulp.

First up is a preprocessor for CSS. We'll be using Sass, so there's a plugin for Gulp called gulp-sass.

```
npm install gulp-sass --save-dev
```

Gulp-sass uses a Sass version developed with the language C, so we won't need to install Ruby. In Sass we'll use partials which are files that are later imported and combined into a main Sass file, so we can better organize our styles, instead of using just one huge CSS file.

However if we generate a CSS file process and want to debug a problem, the problematic line number in the CSS file does not always coincide with the line we have to edit in the SCSS. We saw that with source maps, a kind of file generated together with the CSS file that corresponds each line in the processed CSS with the original Sass file. We have to install gulp-sourcemaps in order to generate sourcemaps for our Sass files

```
npm install gulp-sourcemaps --save-dev
```

Another extremely useful plugin is called auto prefixer. Auto prefixer as the name says automatically adds all the vendor prefixes you need based on the number of routers you want. When using flexbox, it's almost mandatory, because writing the different values for all browsers can be an exhausting task. This functionality was the main reason for me to begin using Gulp.

```
npm install gulp-autoprefixer --save-dev
```

Then there's another plugin that does some work that otherwise would be very time consuming, Imagemin. Imagemin brings together four different programs that optimize GIF, JPG, PNG and SVG images.

```
npm install gulp-imagemin --save-dev
```

Last, we have the automatic reload functionality. Many front end developers are used to updating the page when a file has changed. For me, it's part of my soul at this point. However, you can save yourself this kind of work by installing tools like Live Reload or Browser Sync. We'll use Browser Sync for this project.

```
npm install browser-sync --save-dev
```

Notice that in this case there's no Gulp prefix. That's because browser-sync will work directly with Gulp without the need for a plugin developed by someone else. All packages prefixed with Gulp are Gulp plugins that encapsulate existing software. These are the packages that we'll use for this project.

To get these plugins to work, we need some place to run and configure them. Gulp uses a file called gulpfile.js. So, to create it I'll use my text editor.

```
var gulp = require 'gulp';  
var sass = require 'gulp-sass';  
var sourcemaps = require 'gulp-sourcemaps';
```

```
var autoprefixer = require 'gulp-autoprefixer';  
var imagemin = require 'gulp-imagemin';  
var browserSync = require 'browser-sync'.create();
```