Learn Gulp

Learn gulp by creating a build script to generate a static website like Jekyll.

Jonathan Birkholz

learning with JB

Learn Gulp

Jonathan Birkholz

This book is for sale at http://leanpub.com/learngulp

This version was published on 2015-09-02

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

1. Gulp Series Introduction

Gulp has been a real treat to work with and I wanted to share the ease of which you can use it to quickly create an awesome build.

We are going to be using gulp to build a static content generation site like Jeykll. Obviously it will not be super feature rich, but the basics will be mapped out.

By the end we will have a build script that will run on file change, the content for the pages will be converted from markdown files, and we will have a preview web server that will live reload with these changes.

2. Hello Gulp

Let's start a new node project in our folder and add a package.json.

```
$ npm init
```

```
// /package.json
{
  "name": "learning_gulp",
  "version": "0.0.0",
  "description": "A sample project to learn how Gulp works",
  "main": "index.js",
  "author": "YOUR_NAME_HERE",
  "license": "ISC"
}
```

Time to install `gulp` using `npm`. First globally to access the `gulp` command and then locally for our `package.json`.

```
$ npm install gulp -g
$ npm install gulp --save-dev
```

By default `gulp` looks for a `gulpfile.js` to run. Let's create a simple `gulpfile.js`.

```
// /gulpfile.js
var gulp = require('gulp');

gulp.task('default', [], function() {
  console.log("Hello Gulp! You are mighty fine.");
});
```

In your terminal run the `gulp` command.

```
$ gulp
```

You should see:

```
Using gulpfile ~/YOUR_DIRECTORY/gulpfile.js
Starting 'default'...
Hello Gulp! You are mighty fine.
Finished 'default' after 118 Î¼s
```

Congratulations creating your first gulp build script!

3. Moving Files with Gulp

The first thing we will learn to do with gulp is to move files.

```
/* /contents/styles/some_styles.css */
h1 {
  color: red;
}



/* /contents/styles/more_styles.css */
p {
  font-size: 30px;
}
```

Our project structure should now look like:

```
/node_modules
/contents
  /styles
    more_styles.css
    some_styles.css
gulpfile.js
package.json
```

Update our `gulpfile.js` from the previous section and instruct gulp to move all the files found in the `styles` folder to our `build/styles` folder.

```
// /gulpfile.js
var gulp = require('gulp');

gulp.task('default', [], function() {
  console.log("Moving all files in styles folder");
  gulp.src("contents/styles/**.*")
      .pipe(gulp.dest('build/styles'));
});
```

If we review the code, we can see two main gulp operations.

1. `gulp.src` tells gulp what files to work on with this task.
2. `gulp.dest` tells gulp where to drop off files after this task is finished working on them.

What do we expect will happen when we run `gulp`?

If you guessed the files will be copied and moved to the `build/styles` folder, then give yourself a cookie.

When we run `gulp`, we should see:

```
$ gulp
Using gulpfile ~/YOUR_DIRECTORY/gulpfile.js
Starting 'default'...
Moving all files in styles folder
Finished 'default' after 5.25 ms
```

Our project should now look like:

```
/build
  /styles
    some_styles.css
    more_styles.css
/node_modules
/contents
  /styles
    some_styles.css
    more_styles.css
gulpfile.js
package.json
```

Unimpressed? That just means you totally understand everything and are ready to move on with our great gulp adventure.

4. Concat: Combining multiple files into one with Gulp

Printing `Hello` and moving files is rather boring. Let's do something productive.

When we create websites, we are always trying to deliver the best experience possible. This includes having our webpages displaying fast. Back in the day, this meant having all our styles in one css file.

While this made our webpages load faster, it made maintaining the css file a nightmare!

These days we can use multiple css files for better organization and then concat (meaning merge or combine) the files together into one large file.

We left our project looking like:

```
/build
  /styles
    some_styles.css
    more_styles.css
/node_modules
/contents
  /styles
    some_styles.css
    more_styles.css
gulpfile.js
package.json
```

Right now, we have two separate css files in our `build/styles` folder. We are going to use a gulp plugin to concat all our css files in the `styles` folder.

Gulp contains some basic tasks, but the power of gulp is the customization you can bring into your build process by using plugins.

- For a list of all the gulp plugins available, go to http://gulpjs.com/plugins/

To concat the files together, we will need to install one of these plugins.

```
$ npm install gulp-concat --save-dev
```

- For more information on the `gulp-concat` plugin, visit https://www.npmjs.org/package/gulp-concat/.

We can then update our `default` gulp task to concat the files.

```
// gulpfile.js
var gulp = require('gulp');
var concat = require('gulp-concat');

gulp.task('default', [], function() {
  console.log("Concating and moving all the css files in styles folder");
  gulp.src("contents/styles/**.css")
      .pipe(concat('main.css'))
      .pipe(gulp.dest('build/styles'));
});
```

Couple of things have changed, can you spot them?

First, we had to reference the gulp plugin with:

```
var concat = require('gulp-concat');
```

We chose to label this `concat`. Obviously we could call it anything we want, but `concat` communicates what the plugin does to those reading our build script.

Second, we added another step to our task. In between the `src` and the `pipe(gulp.dest...)` steps, we added `pipe(concat(...))`.

Gulp works by streaming files from one process to another. This allows us to create complex build tasks out of small, simple steps. Composition == winning.

Now run our gulp task:

```
$ gulp
Using gulpfile ~/YOUR_DIRECTORY/gulpfile.js
Starting 'default'...
Moving all the css files in styles folder
Finished 'default' after 6.09 ms
```

Our task will read all the css files in the `styles` folder, combine them into one `main.css` file, and then place that file in the `build/styles` folder.

Our project should now look like:

```
/build
  /styles
    main.css
    more_styles.css
    some_styles.css
/node_modules
/styles
  more_styles.css
  some_styles.css
gulpfile.js
package.json
```

Notice the `more_styles.css` and `some_styles.css` files are still in our build folder. :(

We don’t want those chumps there anymore. In the next chapter we will learn how to get rid of those files.