A Gulp Workshop

Automate Your Web Development Workflow with Gulp

Roy Vanegas

25 June 2016

Introduction

This is a three-hour workshop on the Gulp task runner. By the end of the workshop, you'll have a working, Gulp-driven automation workflow for your HTML/CSS/Sass/JavaScript projects that you can re-use in other projects.

Workshop Links

☞ Repository for these slides

☞ Pre-built Gulp templates

What is Gulp?

Gulp is a stream-based task runner, AKA an automator or build tool, that uses RAM instead of a computer's file system to process data.

It provides an incredibly slim API that allows users to automate many of the tasks that web developers carry out by hand.

What's a Stream?

Streams process data from one task to another in a pipe chain. To do this quickly, Gulp leverages a computer's RAM extensively in order to avoid writing to disk, which can slow data processing.

Installation

The Root User

If you're on a Mac, the `root` user will need to be enabled. Visit this link to learn how. (This doesn't apply to Windows users, and Linux users will already have the `root` user enabled.)

Installation

Node

Install Node from here.

Installation
Gulp

Because Gulp is Node-based, we can now run the following command to install Gulp globally:

```
npm install --global gulp-cli
```

Vertify that it's been installed:

```
which gulp
```

The `package.json` File

Information about a Node project is saved in a manifest file called `package.json`. It's required by your Gulp projects.

The `package.json` File

You may write this file by hand, or have Node create it for you interactively with the `init` option to the `npm` program:

```
npm init
```

The `package.json` File

Saving modules/plugins

For just about every one of your projects, you'll need to use modules, or plugins. (As of this writing, there are 2,483 Gulp plugins.)

Any non-Node module you use needs to be listed in the `package.json` file and included in the `node_modules` folder, which will be explained in a moment.

The `package.json` File

Saving modules/plugins

The easiest way to update `package.json` and to download a module into the `node_modules` folder is to run the `npm install` command with the `--save-dev` flag. Let's download an HTML compressor module:

```
npm install --save-dev gulp-html-minifier
```

The `--save-dev` flag will be explained later.

The `node_modules` Folder

All your modules/plugins **need** to be in a folder called `node_modules` at the root level of your project. As mentioned before, running the `npm install --save-dev MODULE_NAME` command installs your modules into the `node_modules` folder automatically.

The `node_modules` Folder

If your project requires a plugin and it isn't in the `node_modules` folder, or the `node_modules` folder doesn't exist, Gulp will fail.

The Gulp API

Everything in Gulp is driven by a single Node method and 4 Gulp methods:

☞ `pipe()`

☞ `task()`

☞ `src()`

☞ `dest()`

☞ `watch()`

Most Common Gulp CLI Commands

Install Gulp globally

```
npm install -g gulp
```

Most Common Gulp CLI Commands

Install a development dependency

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

Most Common Gulp CLI Commands

Install a production dependency

```
npm install --save gulp
```

Most Common Gulp CLI Commands

Run a gulp task

```
gulp task
```

Most Common Gulp CLI Commands

Run a Gulp task followed by another Gulp task

```
gulp task another_task
```