Bootstrap

— START WITH PACKAGE MANAGERS —

IN A ROCKET

Learn front-end development at rocket speed

BOOTSTRAP
CDN

COMPILED
CSS & JS

SOURCE
FILES

**PACKAGE
MANAGERS**

PACKAGE MANAGERS

NPM

RUBYGEMS

COMPOSER

NUGET

YARN

PACKAGE MANAGERS / NPM / CHEATSHEET

```
$ npm init
```

```
$ npm init -y
```

```
$ npm set init-author-name "John Doe"
```

```
$ npm set init-license "MIT"
```

```
$ npm get init-author-name
```

```
$ npm config delete init-author-name
```

PACKAGE MANAGERS / NPM

```
$ npm init -y
```

PACKAGE MANAGERS / NPM

```
$ npm i bootstrap jquery popper.js -S
```

install

--save

SOURCE: Bootstrap / Documentation / Package Managers.

PACKAGE MANAGERS / NPM

```
$ npm i gulp gulp-sass browser-sync -g
```

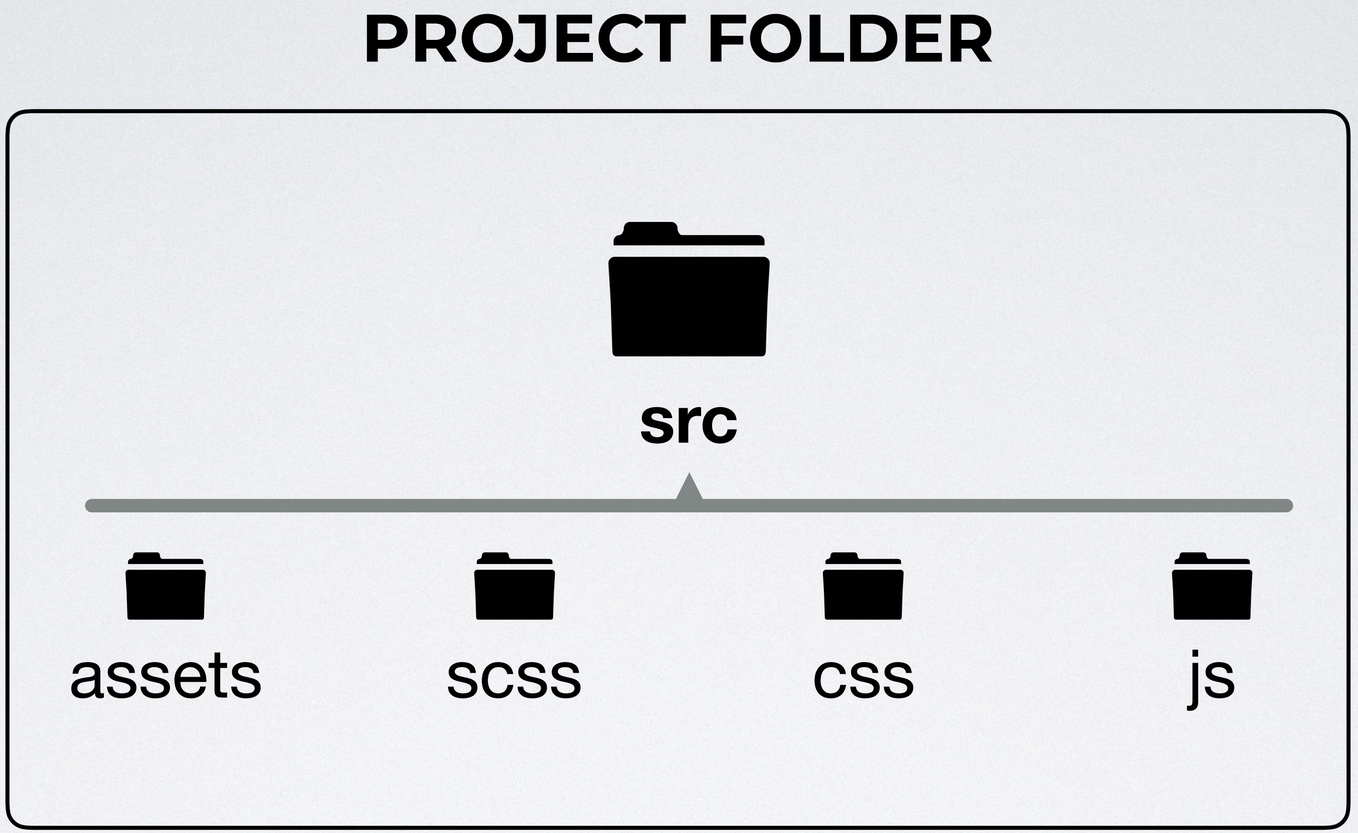
--global

```
$ npm i gulp gulp-sass browser-sync -D
```

--save-dev

SOURCE: Bootstrap / Documentation / Package Managers.

PACKAGE MANAGERS / NPM / PROJECT FOLDERS



PACKAGE MANAGERS / NPM / START TEMPLATE

- src
  - assets
  - scss
  - js
  - css
  - index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Bootstrap 4 Layout</title>
        <meta http-equiv="x-ua-compatible" content="ie=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/styles.css">
    </head>

    <body>
        <script src="js/jquery.min.js"></script>
        <script src="js/popper.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
    </body>
</html>
```

PACKAGE MANAGERS / NPM / SCSS STYLES

- src
  - assets
  - scss
    - **styles.scss**
  - js
  - css

```
$bg-color: red;

body {
  background: $bg-color;
}
```

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html
- **gulpfile.js**

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

// Compile sass into CSS & auto-inject into browsers
gulp.task('sass', function() {
    return gulp.src(['node_modules/bootstrap/scss/bootstrap.scss', 'src/scss/*.scss'])
        .pipe(sass())
        .pipe(gulp.dest("src/css"))
        .pipe(browserSync.stream());
});

// Move the javascript files into our /src/js folder
gulp.task('js', function() {
    return gulp.src(['node_modules/bootstrap/dist/js/bootstrap.min.js', 'node_modules/jquery/dist/jquery.min.js',
'node_modules/popper.js/dist/umd/popper.min.js'])
        .pipe(gulp.dest("src/js"))
        .pipe(browserSync.stream());
});

// Static Server + watching scss/html files
gulp.task('serve', ['sass'], function() {

    browserSync.init({
        server: "./src"
    });

    gulp.watch(['node_modules/bootstrap/scss/bootstrap.scss', 'src/scss/*.scss'], ['sass']);
    gulp.watch("src/*.html").on('change', browserSync.reload);
});

gulp.task('default', ['js','serve']);
```

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html
- **gulpfile.js**

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

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html
- **gulpfile.js**

```
...

// Compile sass into CSS & auto-inject into browsers
gulp.task('sass', function() {
    return gulp.src(['node_modules/bootstrap/scss/bootstrap.scss',
'src/scss/*.scss'])
        .pipe(sass())
        .pipe(gulp.dest("src/css"))
        .pipe(browserSync.stream());
});
```

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html

gulpfile.js

```
...

// Move the javascript files into our /src/js folder
gulp.task('js', function() {
    return gulp.src(['node_modules/bootstrap/dist/js/
bootstrap.min.js', 'node_modules/jquery/dist/
jquery.min.js', 'node_modules/popper.js/dist/umd/
popper.min.js'])
        .pipe(gulp.dest("src/js"))
        .pipe(browserSync.stream());
});
```

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html

gulpfile.js

```
...

// Static Server + watching scss/html files
gulp.task('serve', ['sass'], function() {

    browserSync.init({
        server: "./src"
    });

    gulp.watch(['node_modules/bootstrap/scss/bootstrap.scss',
'src/scss/*.scss'], ['sass']);
    gulp.watch("src/*.html").on('change', browserSync.reload);
});
```

PACKAGE MANAGERS / NPM / GULPFILE

- src
  - assets
  - scss
  - js
  - css
  - index.html
- **gulpfile.js**

```
gulp.task('default', ['js','serve']);
```

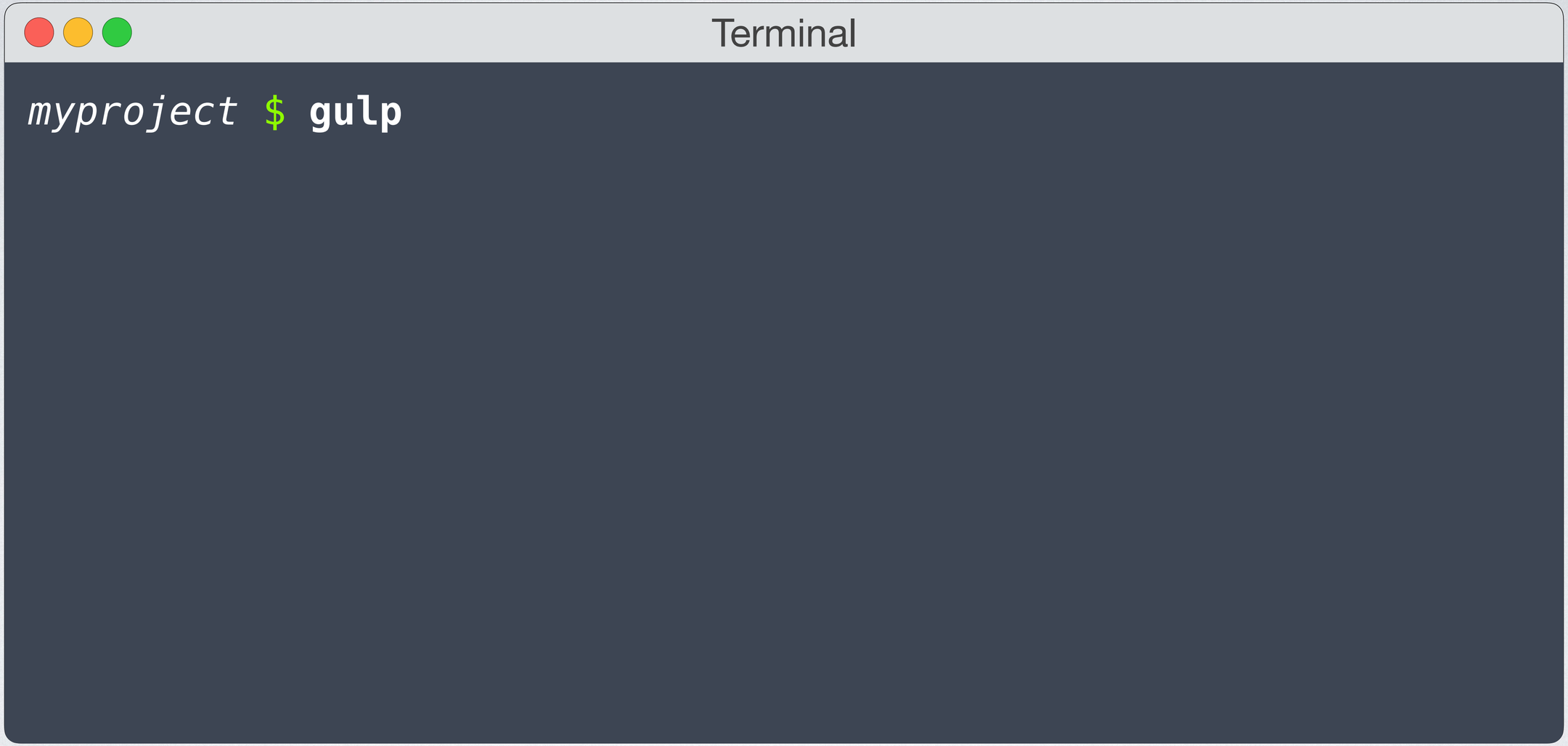
PACKAGE MANAGERS / NPM / GULPFILE

READY TO USE CODE

- src
  - assets
  - scss
  - js
  - css
  - index.html
- **gulpfile.js**

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

// Compile sass into CSS & auto-inject into browsers
gulp.task('sass', function() {
    return gulp.src(['node_modules/bootstrap/scss/bootstrap.scss', 'src/scss/*.scss'])
        .pipe(sass())
        .pipe(gulp.dest("src/css"))
        .pipe(browserSync.stream());
});

// Move the javascript files into our /src/js folder
gulp.task('js', function() {
    return gulp.src(['node_modules/bootstrap/dist/js/bootstrap.min.js', 'node_modules/jquery/dist/jquery.min.js',
'node_modules/popper.js/dist/umd/popper.min.js'])
        .pipe(gulp.dest("src/js"))
        .pipe(browserSync.stream());
});

// Static Server + watching scss/html files
gulp.task('serve', ['sass'], function() {

    browserSync.init({
        server: "./src"
    });

    gulp.watch(['node_modules/bootstrap/scss/bootstrap.scss', 'src/scss/*.scss'], ['sass']);
    gulp.watch("src/*.html").on('change', browserSync.reload);
});

gulp.task('default', ['js','serve']);
```

PACKAGE MANAGERS / NPM / LAUNCH GULP



PACKAGE MANAGERS

NPM

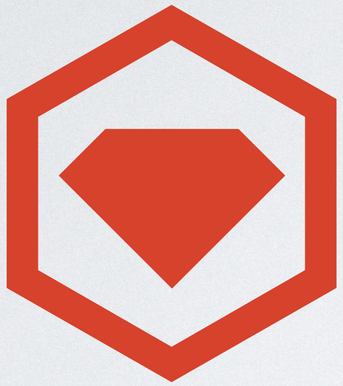
RUBYGEMS

COMPOSER

NUGET

YARN

PACKAGE MANAGERS: RUBYGEMS



Using Bundler (recommended) and RubyGems by adding the following line to your Gemfile:

```
gem 'bootstrap', '~> 4.0.0'
```

If you're not using Bundler, you can install the gem by running:

```
$ gem install bootstrap -v 4.0.0
```

SOURCE: Bootstrap / Documentation / Package Managers.

PACKAGE MANAGERS

NPM

RUBYGEMS

COMPOSER

NUGET

YARN

PACKAGE MANAGERS: COMPOSER

```
$ composer require twbs/bootstrap:4.0.0
```

SOURCE: Bootstrap / Documentation / Package Managers.

PACKAGE MANAGERS

NPM

RUBYGEMS

COMPOSER

NUGET

YARN

PACKAGE MANAGERS: NUGET

```
PM> Install-Package bootstrap
```

```
PM> Install-Package bootstrap.sass
```

SOURCE: Bootstrap / Documentation / Package Managers.

PACKAGE MANAGERS

NPM

RUBYGEMS

COMPOSER

NUGET

YARN

PACKAGE MANAGERS: YARN

```
$ yarn add bootstrap@4.0.0
```

SOURCE: npm / Bootstrap.

Bootstrap

— START WITH PACKAGE MANAGERS —

IN A ROCKET

Learn front-end development at rocket speed