

Grunt

Grunt is a JavaScript-based task runner.

A task runner is an application that strings together tasks you need done during the development process and it runs those tasks for you.

types of tasks Grunt can perform for you

- compile Sass or Less to css
- minify/uglify css or JavaScript files
- concatenate files together
- run tests
- clean up directories and output files
- lint your code

why Grunt?

Grunt is not the only task runner available, nor is it necessarily the best. Grunt, however, is well established, widely used, and has an ecosystem of plugins that totals over 4,000. Ultimately, we're using learning it because it does the jobs we need it to do.

Grunt depends on Node and npm. Node must be installed to execute Grunt and npm to manage and install Grunt plugins and their associated dependencies.

installing Grunt:

```
// install the command-line interface globally
npm install -g grunt-cli

// we'll be installing Grunt on a per-project basis rather than globally
```


package.json

This is where the Grunt plugins will be specified that we're going to be using.

```
{
  "name": "grunt_project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "grunt": "~0.4.5",
    "grunt-contrib-jshint": "^0.11.2",
    "grunt-contrib-watch": "^0.6.1",
    "grunt-mocha": "^0.4.12"
  }
}
```

Create the file initially by using npm init and following the prompts (don't worry about the meta file or repository name warnings.) When done, start installing your dependencies:

```
// in your project directory
npm init
npm install grunt --save-dev
npm install grunt-contrib-jshint --save-dev
npm install grunt-mocha --save-dev
```

Gruntfile.js

Grunt breaks work up into tasks that are specified in this file. Each time you run Grunt, or as we'll see shortly, each time you make a change, Grunt will run all the tasks specified in the default tasks in the Gruntfile, or the tasks you've specified on the command-line

A sample Gruntfile.js. Check documentation for all the possible jshint options.

```
// jshint globals omitted for brevity
module.exports = function (grunt) {
  grunt.initConfig({
    jshint: {
      files: ['GruntFile.js', 'src/**/*.js', 'test/**/*.js', '!**/libs/*.js'],
      options: {
        curly: true,
        eqeqeq: true,
        globals: {
          afterEach: true,
          after: true
        }
      }
    },
    mocha: {
      all: {
        src: ["test/**/*.{html,js}"],
        options: {
          mocha: {
            ignoreLeaks: true
          },
          run: true
        }
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-mocha');
  grunt.registerTask('default', ['jshint', 'mocha']);
};
```

There are 4 key elements to the Gruntfile

```
// 1. a wrapping function assigned to module.exports
module.exports = function (grunt) {

    // 2. a grunt.initConfig method call that passes an object
    grunt.initConfig({
        // task configuration in here
    });

    // 3. one or more calls to the grunt.loadNpmTasks method
    grunt.loadNpmTasks('grunt_plugin_name');

    // 4. zero or more calls to the grunt.registerTask method
    grunt.registerTask('default', ['name of a task specified in initConfig']);
};
```

Running grunt by itself will run the default task, which does the same thing as running grunt default

```
grunt  
  
// equivalent to  
grunt default
```

To run a specific task, use the tasks defined by the registerTask method:

```
grunt <task_name>:<target_name>
```

Running tasks automatically

```
npm install grunt-contrib-watch --save-dev
```

Update Gruntfile.js:

```
module.exports = function (grunt) {
  grunt.initConfig({
    /* ... options ... */
    jshint: {
      files: []
    },
    watch: {
      // reference the files already defined in jshint task
      files: "<%= jshint.files %>",
      tasks: [ "jshint", "mocha" ]
    }
  });
};
```