Exercise (Instructions): Grunt Part 2

Objectives and Outcomes

In this exercise, you will continue to learn to use Grunt, the task runner. You will configure the Grunt file with a set of additional tasks to build your web project. At the end of this exercise, you will be able to:

- Configure a Grunt file with a set of tasks to build your web project from a source.

Copying the Files and Cleaning Up the Dist Folder

- Next you will install the Grunt modules to copy over files to a distribution folder named dist, and clean up the dist folder when needed. To do this, install the following Grunt modules:

```
1   npm install grunt-contrib-copy --save-dev
2   npm install grunt-contrib-clean --save-dev
```

- You will now add the code to perform the copying of files to the dist folder, and cleaning up the dist folder. To do this, add the following code to *Gruntfile.js*. This should be added right after the configuration of the SASS task.:

```
 1  ,
 2
 3      copy: {
 4          html: {
 5              files: [
 6                  {
 7                      //for html
 8                      expand: true,
 9                      dot: true,
10                      cwd: './',
11                      src: ['*.html'],
12                      dest: 'dist'
13                  }]
14          },
15          fonts: {
16              files: [
17                  {
18                      //for font-awesome
19                      expand: true,
20                      dot: true,
21                      cwd: 'node_modules/font-awesome',
22                      src: ['fonts/*.*'],
23                      dest: 'dist'
24                  }]
25          }
26      },
27
28      clean: {
29          build: {
30              src: [ 'dist/']
31          }
32      }
```

- Remember to add the comma after the end of the SASS task.

Compressing and Minifying Images

- Next we install the grunt-contrib-imagemin module and use it to process the images. To install this module type at the prompt:

```
 1  npm install grunt-contrib-imagemin --save-dev
```

- Then, configure the imagemin task as shown below in the Gruntfile:

```
 1  ,
 2      imagemin: {
 3          dynamic: {
 4              files: [{
 5                  expand: true,                    // Enable dynamic expansion
 6                  cwd: './',                       // Src matches are relative to
                     this path
 7                  src: ['img/*.{png,jpg,gif}'],    // Actual patterns to match
 8                  dest: 'dist/'                    // Destination path prefix
 9              }]
10          }
11      }
```

Preparing the Distribution Folder and Files

- We are now going to use the Grunt *usemin* module together with *concat*, *cssmin*, *uglify* and *filerev* to prepare the distribution folder. To do this, install the following Grunt modules:

```
1   npm install grunt-contrib-concat --save-dev
2   npm install grunt-contrib-cssmin --save-dev
3   npm install grunt-contrib-uglify --save-dev
4   npm install grunt-filerev --save-dev
5   npm install grunt-usemin --save-dev
```

- Add the following line to the scripts block in *aboutus.html* and *contactus.html*:

```
1      <script src="js/scripts.js"></script>
```

- Next, update the task configuration within the Gruntfile.js with the following additional code to introduce the new tasks:

```
 1   ,
 2
 3         useminPrepare: {
 4             foo: {
 5                 dest: 'dist',
 6                 src: ['contactus.html','aboutus.html','index.html']
 7             },
 8             options: {
 9                 flow: {
10                     steps: {
11                         css: ['cssmin'],
12                         js:['uglify']
13                     },
14                     post: {
15                         css: [{
16                             name: 'cssmin',
17                             createConfig: function (context, block) {
18                             var generated = context.options.generated;
19                                 generated.options = {
20                                     keepSpecialComments: 0, rebase: false
21                                 };
22                             }
23                         }]
24                     }
25                 }
26             }
27         },
28
29         // Concat
30         concat: {
31             options: {
32                 separator: ';'
33             },
34
35             // dist configuration is provided by useminPrepare
36             dist: {}
37         },
38
39         // Uglify
40         uglify: {
41             // dist configuration is provided by useminPrepare
42             dist: {}
43         },
44
45         cssmin: {
46             dist: {}
47         },
48
49         // Filerev
50         filerev: {
51             options: {
52                 encoding: 'utf8',
53                 algorithm: 'md5',
54                 length: 20
55             },
56
57             release: {
58             // filerev:release hashes(md5) all assets (images, js and css )
59             // in dist directory
60                 files: [{
61                     src: [
62                         'dist/js/*.js',
63                         'dist/css/*.css',
```

```
64                          ]
65                      }]
66                  }
67              },
68
69              // Usemin
70              // Replaces all assets with their revved version in html and css files.
71              // options.assetDirs contains the directories for finding the assets
72              // according to their relative paths
73              usemin: {
74                  html: ['dist/contactus.html','dist/aboutus.html','dist/index.html'],
75                  options: {
76                      assetsDirs: ['dist', 'dist/css','dist/js']
77                  }
78              }
```

- Next, update the jit-grunt configuration as follows, to inform it that useminPrepare task depends on the usemin package:

```
1   require('jit-grunt')(grunt, {
2     useminPrepare: 'grunt-usemin'
3   });
```

- Next, update the Grunt build task as follows:

```
1       grunt.registerTask('build', [
2           'clean',
3           'copy',
4           'imagemin',
5           'useminPrepare',
6           'concat',
7           'cssmin',
8           'uglify',
9           'filerev',
10          'usemin'
11      ]);
```

- Now if you run Grunt, it will create a dist folder with the files structured correctly to be distributed to a server to host your website. To do this, type the following at the prompt:

```
1   grunt build
```

Conclusions

In this exercise you have learnt how to configure a Grunt file to perform several tasks. You were able to build a distribution folder for your web project.

Mark as completed