Back to Week 4 Lessons Prev Next

Exercise (Instructions): Gulp Part 2

Objectives and Outcomes

In this exercise, you will continue to learn to use Gulp. Thereafter you will configure a Gulp file with a set of tasks to build and serve your web project. At the end of this exercise, you will be able to:

- Configure the Gulp file with a set of tasks to build the distribution folder for the web project.

Copying the Files and Cleaning up the Dist Folder

- We will now create the tasks for copying the font files and cleaning up the distribution folder. To do this we will first install the *del* Node module and require it in the Gulp file as follows:

```
npm install del --save-dev
```

```
var ...
    del = require('del'),
    ...
```

- Next, we will add the code for the Clean task and the copyfonts task as follows:

```
// Clean
gulp.task('clean', function() {
    return del(['dist']);
});

gulp.task('copyfonts', function() {
   gulp.src('./node_modules/font-awesome/fonts/**/*.{ttf,woff,eof,svg}*')
   .pipe(gulp.dest('./dist/fonts'));
});
```

Compressing and Minifying Images

- We will now install the *gulp-imagemin* plugin and configure the *imagemin* task. To do this we install the plugin and require it as follows:

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

```
var ...
    imagemin = require('gulp-imagemin'),
    ...
```

- Next, we create the *imagemin* task as follows:

```
// Images
gulp.task('imagemin', function() {
  return gulp.src('img/*.{png,jpg,gif}')
    .pipe(imagemin({ optimizationLevel: 3, progressive: true, interlaced: true }))
    .pipe(gulp.dest('dist/img'));
});
```

Preparing the Distribution Folder and Files

- We now install the gulp-usemin and other related Gulp plugins and require them as follows:

```
npm install gulp-uglify gulp-usemin gulp-rev gulp-clean-css gulp-flatmap gulp-htmlmin --save-dev

```

```
var ...
    uglify = require('gulp-uglify'),
    usemin = require('gulp-usemin'),
    rev = require('gulp-rev'),
    cleanCss = require('gulp-clean-css'),
    flatmap = require('gulp-flatmap'),
    htmlmin = require('gulp-htmlmin');
```

- We configure the usemin and the build task as follows:

```
gulp.task('usemin', function() {
  return gulp.src('./*.html')
  .pipe(flatmap(function(stream, file){
      return stream
        .pipe(usemin({
            css: [ rev() ],
            html: [ function() { return htmlmin({ collapseWhitespace: true })} ],
            js: [ uglify(), rev() ],
            inlinejs: [ uglify() ],
            inlinecss: [ cleanCss(), 'concat' ]
        }))
    }))
    .pipe(gulp.dest('dist/'));
});

gulp.task('build',['clean'], function() {
    gulp.start('copyfonts','imagemin','usemin');
});
```

- Save the Gulp file

Running the Gulp Tasks

- At the command prompt, if you type *gulp build* it will run the build task:

```
     gulp build
```

- Do a Git commit with the message "Gulp Part 2"

Conclusions

In this exercise, you learnt to use Gulp, install Gulp plugins, configure the gulpfile.js and then use Gulp to automate the web development tasks.

Mark as completed