**Grunt**

* Grunt is a task runner. This means that those repetitive tasks we deal with in our daily workflow gets automated.  
  + Linting our JS files
  + Minifying JS files
  + Compiling LESS files
  + Minifying CSS files
  + Watching our files for changes and doing the above tasks
* Let's create project folder, with two sub folders, **src** and **dest**
* package.json file to be created using **npm init** command  
  + {  
     "name": "grunt-getting-started",  
     "version": "0.1.0",  
     "devDependencies": {  
     "grunt": "~0.4.4",  
     "grunt-contrib-jshint": "latest",  
     "jshint-stylish": "latest",  
     "grunt-contrib-uglify": "latest",  
     "grunt-contrib-less": "latest",  
     "grunt-contrib-cssmin": "latest",  
     "grunt-contrib-watch": "latest"  
     }  
    }
* Install all the packages using **npm install** command
* Also install **npm install -g grunt-cli**
* Let's create **Gruntfile.js**
  + // Gruntfile.js  
      
    // our wrapper function (required by grunt and its plugins)  
    // all configuration goes inside this function  
    module.exports = function (grunt) {  
      
     // ===========================================================================  
     // CONFIGURE GRUNT ===========================================================  
     // ===========================================================================  
     grunt.initConfig({  
      
     // get the configuration info from package.json ----------------------------  
     // this way we can use things like name and version (pkg.name)  
     pkg: grunt.file.readJSON('package.json'),  
      
     // all of our configuration will go here  
      
     });  
      
     // ===========================================================================  
     // LOAD GRUNT PLUGINS ========================================================  
     // ===========================================================================  
     // we can only load these if they are in our package.json  
     // make sure you have run npm install so our app can find these  
     grunt.loadNpmTasks('grunt-contrib-jshint');  
     grunt.loadNpmTasks('grunt-contrib-uglify');  
     grunt.loadNpmTasks('grunt-contrib-less');  
     grunt.loadNpmTasks('grunt-contrib-cssmin');  
     grunt.loadNpmTasks('grunt-contrib-watch');  
      
    };
* **grunt.initConfig({})**
  + Place where all of our configuration will go in.
* **grunt.loadNpmTasks()**
  + Loads all the grunt plugins using **grunt.loadNpmTasks()**
* **Linting javascript files**
  + jshint: {  
     options: {  
     reporter: require('jshint-stylish') // use jshint-stylish to make our errors look and read good  
     },  
      
     // when this task is run, lint the Gruntfile and all js files in src  
     build: ['Grunfile.js', 'src/\*\*/\*.js']  
     }
  + grunt jshint
* **Minifying JavaScript files**
  + uglify: {  
     build: {  
     files: {  
     'dest/js/script.min.js': 'src/js/script.js'  
     }  
     }  
     }
  + grunt uglify
* **Minifying multiple files**
  + uglify: {  
     build: {  
     files: {  
     //'dest/js/script.min.js': 'src/js/script.js'  
     'dest/js/script.min.js': ['src/js/script.js', 'src/js/script2.js']  
     }  
     }  
     }
* **Registering a default task**
  + grunt.task.registerTask('default', ['jshint', 'uglify', 'cssmin',]);
  + grunt : to run the tasks
* **Grunt watch**
  + Watches for file changes and perform tasks automatically
  + watch: {  
     css2: {  
     files: ['src/css/\*.css'],  
     tasks: ['cssmin']  
     },  
     js2: {  
     files: ['src/js/\*.js'],  
     tasks: ['jshint', 'uglify']  
     }  
     }

**Entire source code**

// Gruntfile.js  
  
// our wrapper function (required by grunt and its plugins)  
// all configuration goes inside this function  
module.exports = function (grunt) {  
  
 // ===========================================================================  
 // CONFIGURE GRUNT ===========================================================  
 // ===========================================================================  
 grunt.initConfig({  
  
 // get the configuration info from package.json ----------------------------  
 // this way we can use things like name and version (pkg.name)  
 pkg: grunt.file.readJSON('package.json'),  
  
 // all of our configuration will go here  
  
 // configure jshint to validate js files -----------------------------------  
 jshint: {  
 options: {  
 reporter: require('jshint-stylish') // use jshint-stylish to make our errors look and read good  
 },  
  
 // when this task is run, lint the Gruntfile and all js files in src  
 build: ['Grunfile.js', 'src/\*\*/\*.js']  
 },  
  
 uglify: {  
 build: {  
 files: {  
 //'dest/js/script.min.js': 'src/js/script.js'  
 'dest/js/script.min.js': ['src/js/script.js', 'src/js/script2.js']  
 }  
 }  
 },  
  
 cssmin: {  
 build: {  
 files: {  
 //'dest/css/styles.min.js': 'src/css/style.css'  
 'dest/css/styles.min.js': ['src/css/style.css', 'src/css/style2.css']  
 }  
 }  
 },  
  
 watch: {  
 css2: {  
 files: ['src/css/\*.css'],  
 tasks: ['cssmin']  
 },  
 js2: {  
 files: ['src/js/\*.js'],  
 tasks: ['jshint', 'uglify']  
 }  
 }  
  
 });  
  
 grunt.task.registerTask('default', ['jshint', 'uglify', 'cssmin', ]);  
  
 // ===========================================================================  
 // LOAD GRUNT PLUGINS ========================================================  
 // ===========================================================================  
 // we can only load these if they are in our package.json  
 // make sure you have run npm install so our app can find these  
 grunt.loadNpmTasks('grunt-contrib-jshint');  
 grunt.loadNpmTasks('grunt-contrib-uglify');  
 grunt.loadNpmTasks('grunt-contrib-less');  
 grunt.loadNpmTasks('grunt-contrib-cssmin');  
 grunt.loadNpmTasks('grunt-contrib-watch');  
  
};

**GULP**

* Much like Grunt, Gulp is a javascript task runner
* Gulp feels more like a build framework, giving you the tools to create tasks that fit your specific needs.
* Create a package.json file by using command npm init.
* Installation   
  + Install Gulp Globally  
    - npm install --global gulp
  + Install Gulp In devDependencies  
    - npm install --save-dev gulp
  + Create a gulpfile.js  
    - npm install --save-dev gulp-util
* Create **gulpfile.js** file  
  + var gulp = require('gulp'),  
     gutil = require('gulp-util');  
      
    gulp.task('default', function() {  
     return gutil.log('Gulp is up and running!!!');  
    });
* **Gulp functions**
  + **gulp.task :** defines you task and has 3 arguments name of the tasks, dependencies, callback function
  + 2 argument is optional
  + gulp.task('mytask', function() {  
     //do stuff  
    });
  + gulp.task('dependenttask', ['mytask'], function() {  
     //do stuff after 'mytask' is done.  
    });
  + **gulp.src :** points to the files we want to use. It uses .pipe for chaining it’s output into other plugins.
  + **gulp.dest :** points to the output folder we want to write files to.
  + gulp.task('copyHtml', function() {  
     // copy any html files in source/ to public/  
     gulp.src('source/\*.html').pipe(gulp.dest('public'));  
    });
  + **gulp.watch :** watch for file changes and execute the tasks
  + **Install Jshint**
    - **npm install --save-dev jshint gulp-jshint**
    - **npm install --save-dev gulp-jshint jshint-stylish**
* **Sample program**
  + var gulp = require('gulp'),  
     gutil = require('gulp-util');  
      
    gulp.task('default', function() {  
     return gutil.log('Gulp is up and running!!!');  
    });
* **Sample program Jshint**
  + // grab our packages  
    var gulp = require('gulp'),  
     jshint = require('gulp-jshint');  
      
    // define the default task and add the watch task to it  
    gulp.task('default', ['watch']);  
      
    // configure the jshint task  
    gulp.task('jshint', function() {  
     return gulp.src('src/js/\*.js')  
     .pipe(jshint())  
     .pipe(jshint.reporter('jshint-stylish'));  
    });  
      
    // configure which files to watch and what tasks to use on file changes  
    gulp.task('watch', function() {  
     gulp.watch('src/js/\*.js', ['jshint']);  
    });
* **Install minify javascript**
  + npm install --save-dev gulp-concat
  + npm install pump
  + npm install --save-dev gulp-uglify  
    - gulp.task('compress', function (cb) {  
       pump([  
       gulp.src('src/js/\*.js'),  
       gp\_concat('src.min.js'),  
       uglify(),  
       gulp.dest('dest/js/')  
       ], cb);  
      });
* **Install minify css**
  + npm install gulp-css-condense --save-dev
  + gulp.task('compress\_css', function() {  
     return gulp.src('src/css/\*.css')  
     .pipe(cssc())  
     .pipe(gp\_concat('src.min.css'))  
     .pipe(gulp.dest('dest/css/'));  
    });
* **Complete example for GULP**
  + // grab our packages  
    var gulp = require('gulp'),  
     jshint = require('gulp-jshint'),  
     uglify = require('gulp-uglify'),  
     gp\_concat = require('gulp-concat'),  
     cssc = require('gulp-css-condense'),  
     pump = require('pump');  
      
    // define the default task and add the watch task to it  
    gulp.task('default', ['watch']);  
      
    // configure the jshint task  
    gulp.task('jshint', function () {  
     return gulp.src('src/js/\*.js')  
     .pipe(jshint())  
     .pipe(jshint.reporter('jshint-stylish'));  
    });  
      
    gulp.task('compress', function (cb) {  
     pump([  
     gulp.src('src/js/\*.js'),  
     gp\_concat('src.min.js'),  
     uglify(),  
     gulp.dest('dest/js/')  
     ], cb);  
    });  
      
    gulp.task('compress\_css', function() {  
     return gulp.src('src/css/\*.css')  
     .pipe(cssc())  
     .pipe(gp\_concat('src.min.css'))  
     .pipe(gulp.dest('dest/css/'));  
    });  
      
      
    // configure which files to watch and what tasks to use on file changes  
    gulp.task('watch', function () {  
     gulp.watch('src/js/\*.js', ['jshint', 'compress']);  
     gulp.watch('src/css/\*.css', ['compress\_css']);  
    });