Gulp

Task list :

1. Create project
2. Npm init
3. Install gulp
4. Create gulpfile.js
5. Create gulp task
6. Create type script and ts config
7. Create Sass
8. Install plugins
9. Make Default task for run all the tasks
10. Create browser-sync but i didn't see any difference between css and sass
11. Globbing path
12. What is a .pipes
13. Execute project in npm gulp

General analysis

1. Install gulp locally in your project.

```
"devDependencies": {
   "browser-sync": "^2.26.7",
   "gulp": "^4.0.2",
   "gulp-autoprefixer": "^7.0.1",
   "gulp-clean-css": "^4.2.0",
   "gulp-rename": "^2.0.0",
   "gulp-sass": "^4.0.2",
   "gulp-typescript": "^6.0.0-alpha.1",
   "node-sass": "^4.13.1",
   "typescript": "^3.8.2"
 },
```

2. Create a main Gulp execution file.

./gulpfile.js

3. As a first step, create a gulp task that simply launches a message per console. This task will be called "hi." This way gulp will recognize the task when you run “gulp hi”. It is important to

understand that this example command is global. In your case you will point to the gulp executable installed in node_modules.

```
function defaultTask(cb) {
 console.log("hello");
 cb();
}

exports.default = defaultTask
```

4. Compile a script made in typescript (eye! Don't lose focus with typescript! You just have to compile a script that returns JS in ECMAScript 5)

```
gulp.task('typeS', ()=>{
 return tsProject.src()
 .pipe(tsProject())
 .js.pipe(gulp.dest('dist'));
});
```

5. Compile a SASS file to get its corresponding .css. In the same way that in Typescript do not waste time with the example you want to compile, the objective is to know how to compile sass using a Gulp plugin.

```
gulp.task('sass', ()=>{
 return gulp.src('src/**/*.scss')
 .pipe(sass().on('error', sass.logError))
 .pipe(browserSync.stream())
 .pipe(gulp.dest('dist'));
});
```

6. About the result of the previous task; that is, on the .css obtained you will have to apply the corresponding prefixes in the classes that require it. These prefixes are also known as autoprefix. In the “Resources” block we leave you links that can help you to deepen this concept.

```
gulp.task('auto', () =>{
 return gulp.src('dist/**/*.css')
 .pipe(autoprefixer({
     cascade: false
 }))
 .pipe(browserSync.stream())
 .pipe(gulp.dest('dist'))

});
```

7. On the result of the previous task, you will have to apply CSS compression by minifying the content as much as possible to generate the production version.

```
#content{background-color:#7b2a2b;width:50%;height:100px;display:flex;j
ustify-content:center;text-align:center;margin:0
auto}.degradado{background:-moz-linear-gradient(top,#1e5799 0,#2989d8
50%,#207cca 51%,#7db9e8 100%);background:-webkit-gradient(linear,left
top,left
bottom,color-stop(0,#1e5799),color-stop(50%,#2989d8),color-stop(51%,#20
7cca),color-stop(100%,#7db9e8));background:-webkit-linear-gradient(top,
#1e5799 0,#2989d8 50%,#207cca 51%,#7db9e8
100%);background:-o-linear-gradient(top,#1e5799 0,#2989d8 50%,#207cca
51%,#7db9e8 100%);background:-ms-linear-gradient(top,#1e5799 0,#2989d8
50%,#207cca 51%,#7db9e8 100%);background:linear-gradient(to
bottom,#1e5799 0,#2989d8 50%,#207cca 51%,#7db9e8 100%)}
```

8. Rename the output file of the previous task to end the suffix ".min.css" (this suffix will help us understand what is the version of css minified)

./dist/sass/corresponding.min.css

9. Create a task that is invoked by default when executing Gulp (without passing any arguments that specify the name of the task). This is known as the "default task"

```
 gulp.task('default',
gulp.parallel('brows','typeS','sass','auto','minify-css'));

```

10. Create a task that makes use of "browser-sync" observing the change in the sass files.

```
gulp.task('brows',()=>{
 return gulp.src('src/**/*.scss')
 .pipe(sass().on('error', sass.logError))
 .pipe(browserSync.stream())
 .pipe(gulp.dest('dist'));

})
```

11. Create a task that makes use of the "Globbing" concept clearly to work only with files that meet a certain rule. For example, files with extension .scss that are within any directory that is within a main directory called src. (It looks like a pun but it isn't hehe)

```
return gulp.src('src/**/*.scss')
```

12. Understand how .pipe () works in your tasks and what it is used for.

Pipes working with stream we can connected pipe by pipe everytime change something in code

.pipe(do1)
.pipe(do2)
.pipe(do3)
.pipe(save)

13. Configure your project so that you can execute at least one task through npm without using gulp globally or calling the gulp executable installed in node_modules. (investigate how to execute tasks from npm)

```
 gulp.task('default',
gulp.parallel('hi','brows','typeS','sass','auto','minify-css'));
```

Record of lessons learned:

I learned how to create tasks in gulp
What is dev dependency and what is dependency
How to join plugins to main file of gulp
How to convert sass to css