Unit Testing with Jasmine

So, as you all know, next friday we'll be exploring Unit Testing through a small demonstration. JavaScript was the choice. As for the Unit Testing framework: Jasmine!

Here you can find all you need to know to get set up, as well as the source files. Let's get to it.

File Organization

You can find four main folders in this repo.

`DemoJasmine`: Source files for the Jasmine Demonstration;
`DemoJasmine-JQuery`: Source files for the Jasmine-JQuery Demonstration;
`Exercise`: Source files for the exercise you'll be doing!;
`ExerciseSolved`: Our proposed solution for the exercice. You can check it out if you want, it's your call. Or else, just gamble and do it in class;
Each of the Jasmine source folders is organised as follows, with some (but little) variation:

```
root
    SpecRunner.html

    lib
            <jasmine.js and other files>

    spec
            <test specification files>

            helpers
                     <helper files like custom matchers>
    src
            <source files under test>

    coverage
            <code coverage files>
```

Getting Set up

You can clone this git project, or download the zip file. If you don't feel like doing either, just ask us and we'll sent it to you by e-mail.

To test if your ready to go, open the DemoJasmin folder and open SpecRunner.html in your favorite browser. A friendly page should open up, showing all tests passing. Yep, it's that easy. It's just Javascript!

The next steps will allow you to run the code coverage tool. Firstly, if you haven't done this already, install Node.js by going to their website and downloading their installer.

After doing that, grab the command line and move to you **project folder**. Just to be sure, run

```
$ node -v
```

If all went well, the version of Node.js you just installed should be printed. Next, let's get karma to run!

Karma CLI global install

```
$ npm install -g karma-cli
```

If you are using a Mac or Linux machine, don't forget **sudo**

Next, to get everything set up for the code covering tool, run the following commands:

Locally installing jasmine-core, karma-jasmine, phantom-js and karma-coverage

```
$ npm install jasmine-core --save-dev
$ npm install karma-jasmine --save-dev
$ npm install phantomjs karma-phantomjs-launcher --save-dev
$ npm install karma karma-coverage --save-dev
```

We recomend doint this because this way you know what your installing. However, if you want a shortcut, we prepared a package.json file with the dependencies you need. Just run

```
$ npm install
```

Ok, we're ready to go!

How it works

Running tests

The specified tests are run by opening SpecRunner.html in any browser. This HTML file manages all dependencies and necessary JS scripts.

If you prefer the command line, this can also be done, by running the `jasmine` comand. However, you have to use some node.js to manage dependencies from the `src` files.

Code coverage

For code coverage, we will be using Karma. In this example, Karma will fire up our code in a PhantomJS browser and generate the necessary files for examination.

To test it out, move to JasmineDemo directory and type the following in the comand line:

```
karma start
```

Move to the coverage, select PhantomJS and fire up index.html. If everything goes according to plan, a page indicating code coverage should show up.

Read more about it here

Jasmine-JQuery

Jasmine-jquery is a way to include Layout Testing into Jamine Testing through a sieres of custom matchers.

Read more here

Browser compatibility

Jasmin and Karma shoud be supported by all modern browsers. However, for the jQuery fixture to work, we recomend**Firefox**. Chrome and Safary have some security concerns.