Universidade do Porto

Faculdade de Engenharia

FEUP

Static Testing

TVVS

20th, November 2019

André Baptista | Francisco Maria

Exercises:

For the exercises you will be asked to remove the linting errors on **index.js** and make all tests pass.
Each exercise is defined in its own function (**ex1()**, **ex2()** ...) and they have their respective tests (**test1()**, **test2()**, ...).
Once a function has no more linting problems and its test is passing, you can move on to the next exercise. Good luck!

1. For the first exercise, you can follow along the **demo video** found on the repository: https://github.com/carbap/TVVS_StaticTesting

   ```
   "linebreak-style": ["error", "windows"],
   "indent": ["error", "tab"],
   "no-tabs": ["error", { "allowIndentationTabs": true }]
   ```

2. In exercise 2, try to run the tests by doing “node index.js”. You’ll notice that test2 is not passing. The code has no compilation errors, but we get an unexpected behaviour due to a small linting error. Fix the error and the test should now be passing.

3. In exercise 3, similarly to exercise 2, a common mistake is producing an unexpected behaviour that makes test3 fail. Fix that mistake, as well as other linting problems.

4. In exercise 4 there are a few linting problems in the code that you should try to fix. One of them is related to the usage of a "++" operator. If you like using these expressions, you should add the following rule to **.eslintrc.json**:

   "no-plusplus" : ["error", { "allowForLoopAfterthoughts": true }]

5. For exercise 5, start by uncommenting the code in ex5() and the call to test5() at the bottom. If you run the code, you'll see that ex5() never stops. Notice how ESLint detects the problem and try to fix it.

6. For the last exercise, we want to demonstrate one way in which static analysis can prevent security vulnerabilities.
   Go ahead and open **index.html** and **script.js**. Notice that ESLint is disabled in this script.
   There is a major security issue in this code that might not be obvious at first sight. Try to exploit this issue. After identifying the problem, enable ESLint and you can conclude that it would have warned you about the potentially harmful code.

   (hint: is the script safe against code injection?)