Lab 07 - Testing your code

Script Languages

Learning goals

1. Know what a unit test is.
2. Write a test on a function returning a value or an exception.
3. Know what is a TDD (Test Driven Design).

Exercises

1. Preparation

In this lab you will create code to test Python function. Please, try not to read ahead a whole instruction, but do each task one by one.

1. Install and read about `pytest` module.

   https://docs.pytest.org/

2. Create a simple class to store HTTP request, containing: request type, resource path and HTTP protocol type.

2. Test driven development

1. Write a Python function:

```
def reqstr2obj(request_string):
    pass
```

2. Write a test #1 to check if the function raises an exception `TypeError` if `request_string` is not of the string type.
3. Change the function to pass the test. Does the function have to return an object to pass this test? Does the function have to return anything?
4. Write a test #2 to check if the function returns an object of an HTTP request class (developed individually in task 1.1) if called with an argument: "GET / HTTP1.1". Does the function have to return an object with attributes set correctly to pass this test?
5. Write a test #3 to check if the function called with an argument "GET / HTTP1.1" returns an object of HTTP request class attributes set accordingly to: GET, /, HTTP1.1. What would happen if you end the function with a statement: return

CLASS("GET", "/", "HTTP1.1") (where CLASS is a class developed in task 1.1)? Will such a function pass this test?

6. Write a test #4 to check if the function returns an object relevant to the argument, not only the one from test #3.

7. Write a test #5 to check if the function returns `None` if a request string does not consist of three words separated by a single space.

8. Write a test #6 to check if the function raises exception `BadRequestTypeError` if called with string that uses illegal request type (e.g. "DOWNLOAD /movie.mp4 HTTP1.1”).

9. Write a test #7 to check if the function raises exception `BadHTTPVersion` if called with a string that contains string other then: `HTTP1.0, HTTP1.1` or `HTTP2.0.`

10. Write a test #8 to check if the function raises exception `ValueError`, containing a text “Path must start with /” if the path does not start with the slash (“/”) character.

11. Remove the function, and write it from scratch, step by step again to pass one more test with each run. Be prepared to remove the function on a request of your teacher, and to rewrite it from scratch to pass the particular test ONLY.