

Lab 4 Notes

- In the first challenge we have the result of report to be work we should add a command and it is there in the screenshot:

```
(tp4-env) PS C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4> $env:PYTHONPATH = (Get-Location).Path
(tp4-env) PS C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4> pytest tests/model/test_module.py
===== test session starts =====
platform win32 -- Python 3.12.7, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4
collected 2 items

tests\model\test_module.py ..                                           [100%]

===== 2 passed in 0.06s =====
(tp4-env) PS C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4>
```

- in challenge 2.1

```
(tp4-env) PS C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4> pytest tests/model/test_2_row_list.py
===== test session starts =====
platform win32 -- Python 3.12.7, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4
collected 3 items

tests\model\test_2_row_list.py .FF                                     [100%]

===== FAILURES =====
test_for_missing_area
>
def test_for_missing_area():
> assert row_to_list("\t293,410") is None
E       AssertionError: assert ['293,410'] is None
E       + where ['293,410'] = row_to_list("\t293,410")
tests\model\test_2_row_list.py:8: AssertionError
test_for_missing_tab
>
def test_for_missing_tab():
> assert row_to_list("\t1,463238,765\n") is None
E       AssertionError: assert ['1,463238,765'] is None
E       + where ['1,463238,765'] = row_to_list("\t1,463238,765\n")
tests\model\test_2_row_list.py:11: AssertionError
===== short test summary info =====
FAILED tests/model/test_2_row_list.py::test_for_missing_area - AssertionError: assert ['293,410'] is None
FAILED tests/model/test_2_row_list.py::test_for_missing_tab - AssertionError: assert ['1,463238,765'] is None
===== 2 failed, 1 passed in 0.11s =====
(tp4-env) PS C:\Users\hafsa\OneDrive\Documents\SEDS\SEDS_Lab4\tp4>
```

- textblob is a library that analyse if a text is sentiment positive (happyness..etc) ou négative(disopointed...),it provides an easy way to calculate the "polarity" (sentiment score) of text
- Git thinks of its data as a set of snapshots (`commit`) of a miniature filesystem.Every time a project state is saved (committed), Git stores a reference to that snapshot.
- A snapshot is identified by **SHA-1 hash**
- Git assigns references for hashes , For example:
 - " **master**" (or main) ⇒ points to the latest commit in the main branch of

development.

○ "

Head" ⇒ points to "where we currently are".

- when we create a new commit the head will point in master however head~1 will point in the old commit , to see the history of old commits we do : `git diff head~1` , and to change my place to head~1 (bcz as we said head is where we point now) we write : `git checkout head~1`
- **HEAD** : points in where we are now
- **HEAD~1**: to go in the commit before where we are
- **master (or any name of the branch)**: points in the last commit in that branch
- **master** branch is the **default** branch.
- **main branch** is the branch that we work in it now.
-