Exercise 1

Clone repository:

```
git clone https://github.com/BI-DS/GRA-4152
```

a. Explore the version history by visualizing it as a graph

```
git log --graph
* commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498 (HEAD -> master, origin/master,
origin/HEAD)
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Wed Aug 31 09:59:14 2022 +0200
|
|     added honor code
|
* commit a610fc6f9cdf3dd3e84e42e8e6ac168a73ab3a28
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Mon Aug 29 09:33:29 2022 +0200
|
|     adding 1 asnyc exercise from lecture 1
|
* commit 0f6036be424d789df67656a3be50cec8848f2d4f
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Mon Aug 29 09:31:42 2022 +0200
|
|     adding 1 asnyc exercise from lecture 1
|
* commit ae45ae7891f40dab569153b9decaaedc71601ff5
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Mon Aug 29 08:51:21 2022 +0200
|
|     removing git_exercise.py file
|
* commit 84ed53d35856ac4c6a59a8de7853e2d723b44a39
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Fri Aug 26 08:56:46 2022 +0200
|
|     adding file for git exercise in lecture 2
|
* commit 37ac00f151ef14e398bf6ab3c3078a799d974775
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Mon Aug 22 11:57:28 2022 +0200
|
|     adding some py files for lecture 1
|
* commit 0fb7842d8311144c4d1941b8e9d828059e11c500
| Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
| Date:   Mon Aug 22 08:50:07 2022 +0200
|
|     adding instructions for UML
|
* commit 20b88515dc1668bed7942359e3ad183f51961f62
  Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
  Date:   Fri Aug 19 08:48:04 2022 +0200

      initial commit
```

b. Last modification of README.md

```
git log README.md
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498 (HEAD -> master, origin/master,
origin/HEAD)
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code

commit 0fb7842d8311144c4d1941b8e9d828059e11c500
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Mon Aug 22 08:50:07 2022 +0200

    adding instructions for UML

commit 20b88515dc1668bed7942359e3ad183f51961f62
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Fri Aug 19 08:48:04 2022 +0200

    initial commit
```

As we see highlighted in the code, the last modification took place 31.aug 2022, the honor code was added to the file.

c. The commit message associated with the last modification of the file:

```
git blame README.md
^20b8851 (rogelioandrade 2022-08-19 08:48:04 +0200  1) # GRA-4152
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  2) This repository contains different
materials used throughout the course, e.g. examples shown in lectures, suggestd solutions
for homework, problems discussed in tutorial sessions, etc. You should follow this
repository frequently, as different materials will become available as we cover th
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  3)
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  4) ## Packages
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  5) Unified Modeling Language (UML) is a
tool to visualize the design, or architecture, of (complex) software systems. Just like
classes in OOP. We can generate UML diagrams for `Python` classes using the library
`pylint`, which uses `graphviz` to generate `png` or `pdf` files showing the architecture
of a given class.
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  6) ```bash
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  7) pip install pylint
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  8) sudo apt install graphviz
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  9) pyreverse -o png <your code>.py
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 10) ```
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 11)
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 12) ## Honor Code
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 13) You are free to form study groups
and may discuss homework in groups. However, each student must write down the solutions and
code from scratch independently and must understand the solution well enough. It is a honor
code violation to copy, refer to, or look at written or code solutions from a previous year
or solutions posted online (inspired by the Stanford Honor Code).
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 14)
```

The abbreviated hash of the last commit is: 71f261f8. To read the message associated with the last commit we use git show and specify the format and properties of the hash we do want to read. In the code below "-s" allows to suppress the changes made in the file, while "—format = %s" allows to get only commit message as output.

```
git show -s --format=%s 71f261f8
added honor code
```
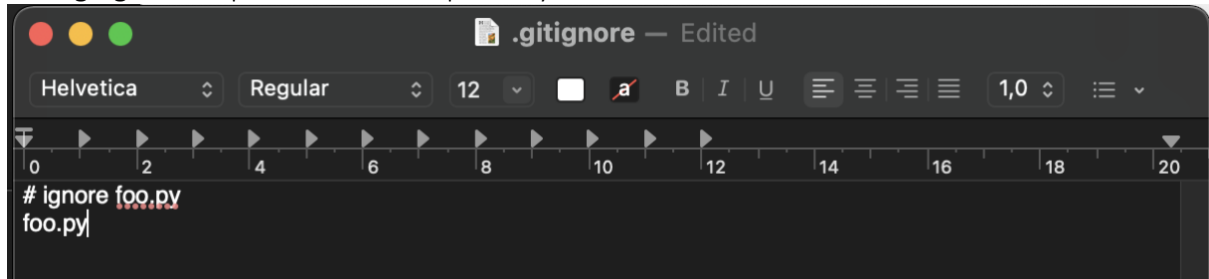
Exercise 2

```
git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .DS_Store

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GRA-4152/

        foo.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Add .gitignore to portfolio code repository and exclude files and folders.

After adding the file:

```
git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .DS_Store

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GRA-4152/

no changes added to commit (use "git add" and/or "git commit -a")
```

Exercise 3

```
1. git clone https://github.com/cantaro86/Financial-Models-Numerical-Methods
2. cd "/Users/          /Documents/skole/MSc in Quantitative Finance/3rd semester/GRA4152/
Financial-Models-Numerical-Methods "
```

After modifying *requierements.txt* from the repository we check status and run git stash:

```
git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   requirements.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")

git stash
Saved working directory and index state WIP on master: 7a7f2b2 Create FUNDING.yml
```

When we run **git stash**, we basically store temporarily changes on a local disc without committing them.

After running **git log --all** we get the list of all commits and corresponding hashes and messages. Furthermore, on the first line, we can find the hash for the stash we run previously.

```
git log --all --oneline
919a2d1 (refs/stash) WIP on master: 7a7f2b2 Create FUNDING.yml
4ec7a1e index on master: 7a7f2b2 Create FUNDING.yml
7a7f2b2 (HEAD -> master, origin/master, origin/HEAD) Create FUNDING.yml
6514fda typo notebook OU
9876dad typo
88f3fa3 Correction time to exit strip
...
```

```
git stash pop
Dropped refs/stash@{0} (919a2d1b1813116c6ea92e6ba3b64a4f89e14937)
```

The main use of the git stash is when the situation requires to switch from task to another immediately. Instead of committing the modifications that are not completed and switching master one can just stash the modifications and jump on the other project.

To list all the stashes, we can use:

```
git stash list
stash@{0}: WIP on master: 7a7f2b2 Create FUNDING.yml
```

In the second line we see the stash id. To drop the stash we have to run the following code:

```
git stash drop stash@{0}
Dropped stash@{0} (75374b50ceaaeaff7a6d2549afccb42f336f1ea5)
```

After we modified and commit the file that was stashed previously, we will get the conflict:

```
git stash pop
Auto-merging requirements.txt
CONFLICT (content): Merge conflict in requirements.txt
The stash entry is kept in case you need it again.
```

If we open the file we can observe the following:

```
<<<<<<< Updated upstream
blablabla
=======
pypypyton
>>>>>>> Stashed changes
```

Exercise 4

Create a new branch:

```
git branch my_test_branch
git log --oneline
6b70ef3 (HEAD -> master, my_test_branch) added .gitignore to repo
d27a8c8 (origin/master) initial commit
6b43af2 initial commit
```

As we see in the highlighted line, the arrow points at the current branch. Let us switch the branch:

```
git checkout my_test_branch
M         .DS_Store
Switched to branch 'my_test_branch'

git log --oneline
6b70ef3 (HEAD -> my_test_branch, master) added .gitignore to repo

d27a8c8 (origin/master) initial commit
6b43af2 initial commit
```

To switch back we run:

```
git checkout master
M         .DS_Store
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
```

Last part of the task:

```
git checkout my_test_branch
M         .DS_Store
Switched to branch 'my_test_branch'

git log --oneline
8726733 (HEAD -> my_test_branch, master) document we are going to use for modifications
6b70ef3 added .gitignore to repo
d27a8c8 (origin/master) initial commit
6b43af2 initial commit

git status
On branch my_test_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .DS_Store
        modified:   test_doc.rtf

git commit -a -m "added comments"
[my_test_branch edfb791] added comments
 2 files changed, 2 insertions(+), 1 deletion(-)

git checkout master
git merge my_test_branch
Updating 8726733..edfb791
Fast-forward
 .DS_Store      | Bin 6148 -> 10244 bytes
 test_doc.rtf |    3 ++-
 2 files changed, 2 insertions(+), 1 deletion(-)
```

Exercise 5

Fork the class repo:

And clone it:

```
git clone https://github.com/S1814771/GRA-4152
Cloning into 'GRA-4152'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 32 (delta 8), reused 27 (delta 3), pack-reused 0
Receiving objects: 100% (32/32), 4.43 KiB | 4.43 MiB/s, done.
Resolving deltas: 100% (8/8), done.

cd "/Users/jakovsei/Documents/skole/MSc in Quantitative Finance/3rd semester/GRA4152/GRA-4152"
```

```
git checkout -b request master
git remote add upstream https://github.com/BI-DS/GRA-4152
git add S1814771.txt
git commit -m "Sending a pull request to add a txt-file"
[request 2782b2f] Sending a pull request to add a txt-file
 1 file changed, 1 insertion(+)
 create mode 100644 S1814771.txt

git push -u origin request
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'request' on GitHub by visiting:
remote:      https://github.com/S1814771/GRA-4152/pull/new/request
remote:
To https://github.com/S1814771/GRA-4152
 * [new branch]      request -> request
Branch 'request' set up to track remote branch 'request' from 'origin'.
```