1. Clone the repository for the class: https://github.com/BI-DS/GRA-4152

cd C:/Users/calle/Desktop/OOP/github
git clone https://github.com/BI-DS/GRA-4152.git

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

cd ./GRA-4152
git log –graph

```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/github/GRA-4152 (master)
$ 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
```

b. When was the last time README.md was modified? (Hint: use git log with an argument).

git log README.md

```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/github/GRA-4152 (master)
$ 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
```

Aug 31st .

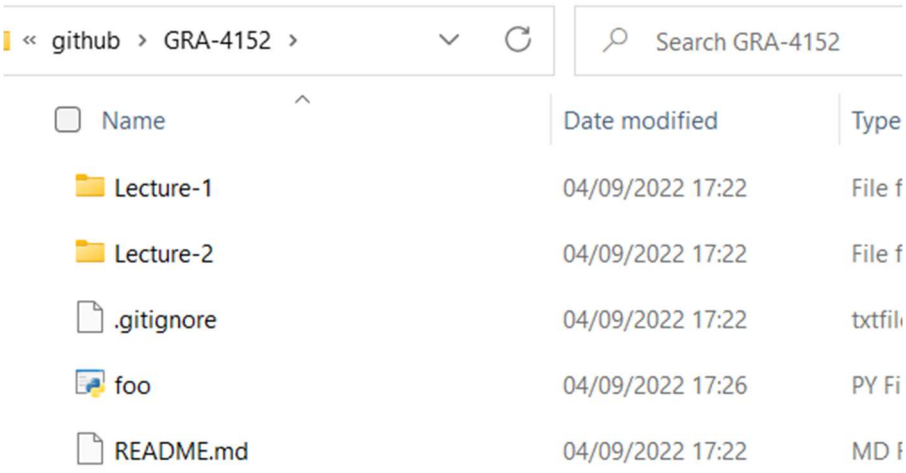
c. What was the commit message associated with the last modification to the README.md? (Hint: use git blame and git show).

git blame README.md
git show 71f261f8

"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)."

2. One common mistake when learning Git is to commit large files that should not be managed by Git or adding sensitive information. Add a .gitignore file to your portfolio code repository (https://github.com/ SXXXXXXX/GRA4152) and exclude files and/or folders. You might need to create a foo.py file to be excluded.

echo "# GRA4152" >> foo.py
git add .gitignore



3. Clone some repository from GitHub and modify one of its existing files. What happens when you type git stash? What do you see when running git log --all --oneline? Run git stash pop to undo what you did with git stash. In what scenario might this be useful? List your current stashes and delete them with git stash drop <stash_id>

git clone https://github.com/bitcoin/bitcoin.git
(modify .gitignore)
cd ./bitcoin
git stash

```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/github/GRA-4152/bitcoin (master)
$ git stash
Saved working directory and index state WIP on master: 604015ac7 Merge bitcoin/bitcoin#25914: test: Fix intermittent issue in p2p_leak.py
```

git log --all – oneline
(nothing)
git stash pop

```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/github/GRA-4152/bitcoin (master)
$ git log --all -- oneline

calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/github/GRA-4152/bitcoin (master)
$ git stash pop
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:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (4a21c628a5df2c3f52e88d31e6bf7a46876e5a4e)
```

(useful if don't want to commit local changes)
git stash list
git stash drop

Now, modify a file and stash changes. Make a new modification to the same file, but this time commit those changes. What happens if you type git stash pop and open the file that you have modified? What do you see in the file?

git stash
git add .gitignore
git commit
(file changed even if git stash pop)

4. Create a new branch in your class repository (https://github.com/ SXXXXXXX/GRA4152) and call it my_test_branch. Explore both branches, by switching back and forth.

git branch my_test_branch
git checkout my_test_branch

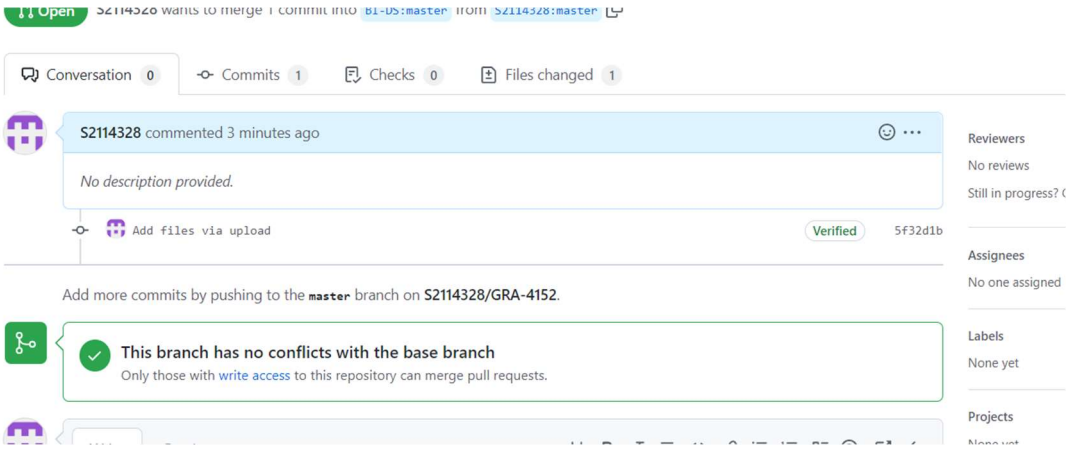
```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/GRA4152 (main)
$ git checkout my_test_branch
Switched to branch 'my_test_branch'
```

echo «# GRA4152» >> foo.py
(add line of text to foo)
git add foo.py
git commit -m "nline"
git checkout master
git merge my_test_branch

```
calle@LAPTOP-RPLGQ6HD MINGW64 ~/Desktop/OOP/GRA4152 (master)
$ git merge my_test_branch
Updating 08debe7..1649865
Fast-forward
 foo.py | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 foo.py
```

5. Fork the class repository (https://github.com/BI-DS/GRA-4152) and clone it, so you can contribute to its development. Send me a pull request to add a text file with your student id, i.e. SXXXXXXX.txt

git clone https://github.com/S2114328/GRA-4152.git


Open
S2114328 wants to merge 1 commit into BI-DS:master from S2114328:master
Conversation 0
Commits 1
Checks 0
Files changed 1
S2114328 commented 3 minutes ago
No description provided.
Add files via upload
Verified
5f32d1b
Add more commits by pushing to the master branch on S2114328/GRA-4152.
This branch has no conflicts with the base branch
Only those with write access to this repository can merge pull requests.
Reviewers
No reviews
Still in progress?
Assignees
No one assigned
Labels
None yet
Projects