Homework LIMONIER Joris

1. cd vip_test
   git pull

2. git tag start_L.4

3. touch 1234.stderr 1234.stdout 432.stderr 432.stdout

4. Copy-paste from a github .gitignore template for python. Then I add the following lines:
   *.stderror
   *.stdout
   Then I execute the following commands:
   git add .
   git commit -m "" *(I had done this during the previous lecture and I had forgotten to add this commit so it does not appear in my commit list, but I would write something like git commit -m "add stdout and stderr files to .gitignore")*
   git push

5. Done.

6. git tag addignore

7. git add .
   git commit -m "modify gitignore from lottery"
   git push

8. Done

9. I use mv <name>.<ext> <name>_1.<ext>
   Then:
   git add .
   git commit -m "add '_1' to all .yml and .ipynb files"
   git push

10. I use mv <name>_1.<ext> <name>.<ext>
    Then *(I had forgotten to do it during last lecture, but here is what I would have done)*:
    git add .
    git commit -m "remove'_1' to all .yml and .ipynb files"
    git push

11. Done

12. git checkout addignore .gitignore

13. git status

14. git add .
    git commit -m "add untracked files" *(all my files were already tracked so there was nothing to commit"*

git push

15. Done

16. This didn't work for me (*I saw your message on Slack only after I was done with the*

17. git config --list

18. I cannot because of question 16

19. git checkout -b my_branch

20. rm -r *.stderr *.stdout *.py.save cycle_gan/ CycleGAN.ipynb data datasets/ docs/ environment.yml imgs/ LICENSE models/ options/ pix2pix.ipynb requirements.txt scripts/ util/ VP.jpeg
    git add .
    git commit -m "keep only *.py, test_file and README.md"
    git push -u origin my_branch

21. git add file_from_my_branch
    git commit -m "add file_from_my_branch"
    git push

22. Done

23. git checkout main

24. Using GitHub CLI:
    gh pr create
    → *then merge from GitHub website*

25. git branch -d my_branch

26. git checkout -b new_branch

27. nano README.md
    this line was added from new_branch
    [Ctrl+X]
    [Y]
    [Enter]
    git add .
    git commit -m "add line to README"
    git push -u origin new_branch

28. git checkout main

29. nano README.md
    this line was added from new_branch
    [Ctrl+X]
    [Y]
    [Enter]
    git add .
    git commit -m "add line to README (from main)"
    git push

      → *Gives an error because remote contains work that I do not have locally*

30. Using GitHub CLI:
    gh pr create
    → *then try to merge from GitHub website*
    → Since the README's from both branch have new changes, there are conflicts that need to be solved

31. Since it is bad practice to push directly to main branch, I deleted the changes made in main.

32. I cannot because of question 16

33. git checkout start_L.4

34. git switch -c old_version

35. git push -u origin old_version

36. git status
    ls

37. git checkout start_L.4 environment.yml
    (this outputs "Updated 0 paths from d4e30d2")

38. git status
    ls
    (no difference)

39. git add .
    git commit -m "tried to restore environment.yml"
    git push

40. I cannot because of question 16

41. git reset --hard gitignore
    fatal: ambiguous argument 'gitignore': unknown revision or path not in the working tree.

42. I cannot because of question 16

43. I cannot because of question 16

44. Done