Practical work for lesson 4 of course Basic tools

This list of exercises is designed only to help you learn how to work with GitHub. So, if you have any questions about the exercises (or you need some help), please **do not hesitate to ask me!**

Please make a short and structured report of used commands (via copy-paste), outputs from terminal and short answers (if required) for each exercise and sent to my email victoriya.kashtanova@inria.fr (or in the slack private messages) in the end of session.

Note: During this session make only UNDERSTANDABLE git commits (with the descriptions of your actions).

Exercises

Gitignore:

1. Go to **vip_test** directory on your Linux machine and **pull** changes from GitHub.
2. Make a tag with command **git tag start_L.4**
3. Create 4 zero-size files : **1234.stderr, 1234.stdout, 432.stderr, 432.stdout**
4. Create standard for python repository **.gitignore** file with additional ignoring for types of files from ex.3; add, commit and push it to remote.
5. Check your remote (on GitHub site) for changes.
6. Make a tag **addignore**
7. Add to your **.gitignore** all types of files and directories which you get from the lottery; add, commit and push it to remote.
8. Check your remote (on GitHub site) for changes.
9. Rename all files and directories from lottery to **<name>_1**; add, commit and push it to remote.
10. Rename again all previously changed files to **<name>** (without "_1"); add, commit and push it to remote.
11. Check your remote (on GitHub site) for changes.
12. Rollback your **.gitignore** file to the version from tag **addignore** with command **git checkout addignore .gitignore**
13. Check your **git status**.
14. Add and commit untracked files and directories, push all to remote.
15. Check your remote (on GitHub site) for changes.

Branching:

16. Add an alias **git hist** with command :
    ```
    git config --global alias.hist=log --pretty=format: '%h %ad | %s%d [%an]' --graph --date=short
    ```
17. Check your **git config --list** to see list of all your global git configurations.
18. Run **git hist** to get a fancy graph of all your commits.

19. Create a new branch **my_branch** for your **vip_test** directory and make it your working brunch.
20. Delete all files and directories except **README.md**, **test_file** and **.py** files; add, commit and push it to remote.
21. Create file **file_from_my_branch**, add, commit and push it to remote.
22. Check this branch in your remote (on GitHub site).
23. Go back to **master** branch.
24. **Merge** your **master** with **my_branch**
25. Delete **my_branch**
26. Create a new branch **new_branch** for your **vip_test** directory and make it your working brunch.
27. Add line **"This line was added from new_branch"** to your **README.md**; add, commit and push it to remote.
28. Go back to **master** branch.
29. Add line **"This line was added from master"** to your **README.md**; add, commit and push it to remote.
30. Try to **merge** your **master** with **new_branch**
31. Resolve the conflict; add, commit and push it to remote.
32. Check **git hist**

Old version:

33. Get previous version of your repository via command **git checkout start_L.4**
34. Create a new branch **old_version** with the **old version of your vip_test** repository (use command proposed by git), **push** changes.
35. Go back to your "new" version via **git checkout master**
36. Check **git status** and files in your repository via **ls**.
37. Try to restore file **environment.yml** via **git checkout start_L.4 environment.yml**
38. Check **git status** and files in your repository via **ls**.
39. Add and commit untracked file, push it to remote.
40. Check **git hist**

41. Rollback all your repository to the **addignore** version via command
git reset --hard addignore
42. Check **git hist** (it is your new current history of commits).
43. To see all history of commits you can use **git hist --all**
44. (if you want) You can restore your "new" version via command
git checkout <hash of you last commit from git hist --all>