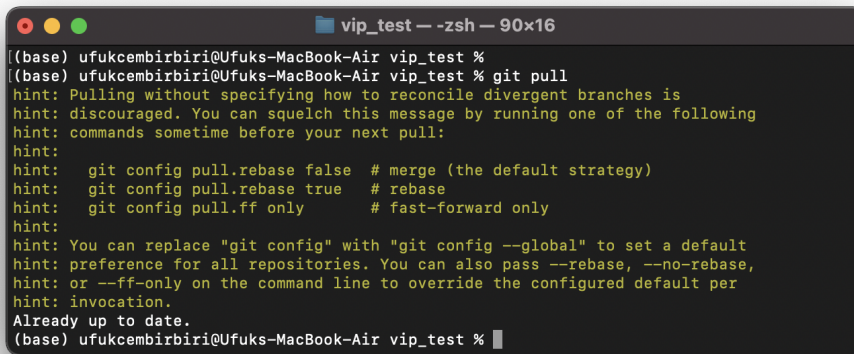


BASIC TOOLS - GIT EXERCISES-2

UFUK CEM BIRBIRI

Gitignore:

1- git pull

A terminal window titled 'vip_test - zsh - 90x16' showing a git pull command and its output. The output includes several hints about reconciling divergent branches and configuring pull behavior.

```
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %  
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git pull  
hint: Pulling without specifying how to reconcile divergent branches is  
hint: discouraged. You can squelch this message by running one of the following  
hint: commands sometime before your next pull:  
hint:  
hint:   git config pull.rebase false  # merge (the default strategy)  
hint:   git config pull.rebase true   # rebase  
hint:   git config pull.ff only       # fast-forward only  
hint:  
hint: You can replace "git config" with "git config --global" to set a default  
hint: preference for all repositories. You can also pass --rebase, --no-rebase,  
hint: or --ff-only on the command line to override the configured default per  
hint: invocation.  
Already up to date.  
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

2-git tag start_L.4

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

4-

.gitignore

echo "*.stderr" >> .gitignore

echo "*.stdout" >> .gitignore

git add .gitignore

git commit -m "commit for .gitignore"

git push

```
vip_test — zsh — 90x32
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % touch .gitignore
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % echo "*.stderr" >> .gitignore
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % echo "*.stdout" >> .gitignore
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % cat .gitignore
*.stderr
*.stdout
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % ls
1234.stderr      README.md        models
1234.stdout      data             options.lua
432.stderr       datasets         pretrained_models
432.stdout       examples         test.lua
CycleGAN         file_trial       train.lua
LICENSE          imgs            util
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git add .gitignore
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git commit -m "commit for .gitignore"
[master 82d3592] commit for .gitignore
 1 file changed, 2 insertions(+)
   create mode 100644 .gitignore
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 290 bytes | 290.00 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.
To https://github.com/CemBirbiri/vip_test.git
 67b87ee..82d3592  master -> master
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

5- git tag addignore

7-

```
echo "**/scripts/" >> .gitignore
echo "**/imgs/" >> .gitignore
echo "*.yaml" >> .gitignore
echo "*.ipynb" >> .gitignore
git add .gitignore
git commit -m "2- commit for gitignore"
git push
```

9- I wrote following commands:

```
mkdir scripts_1
mkdir imgs_1
touch lottery_1.yaml
touch lottery_1.ipynb
git add scripts_1 lottery_1.yaml lottery_1.ipynb imgs_1
```

Then a warning occurred in the terminal:

The following paths are ignored by one of your .gitignore files:

lottery_1.ipynb

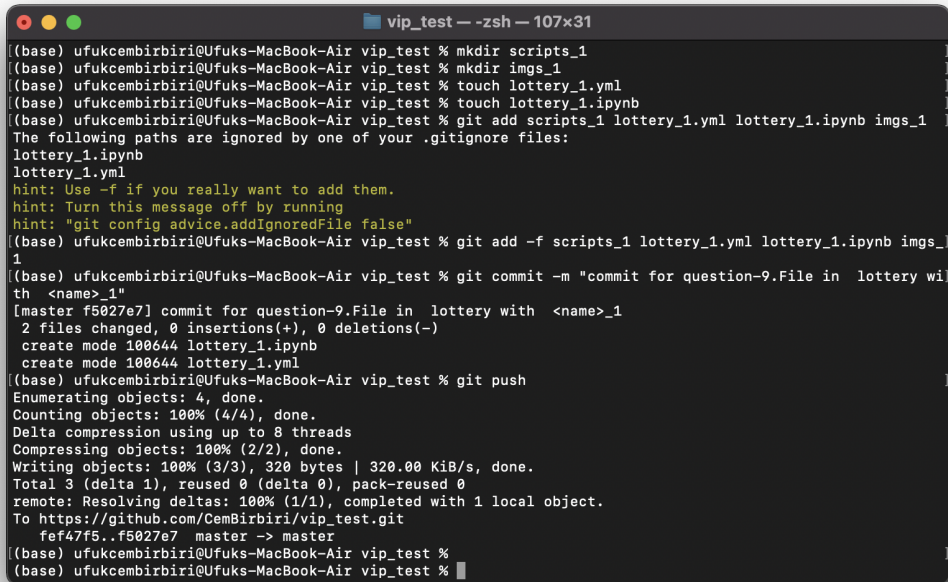
lottery_1.yaml

hint: Use -f if you really want to add them.

So I used following commands:

```
git add -f scripts_1 lottery_1.yaml lottery_1.ipynb imgs_1
```

git commit -m "commit for question-9.File in lottery with <name>_1"
git push



```
vip_test — zsh — 107x31
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % mkdir scripts_1
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % mkdir imgs_1
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % touch lottery_1.yml
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % touch lottery_1.ipynb
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git add scripts_1 lottery_1.yml lottery_1.ipynb imgs_1
The following paths are ignored by one of your .gitignore files:
lottery_1.ipynb
lottery_1.yml
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git add -f scripts_1 lottery_1.yml lottery_1.ipynb imgs_1
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git commit -m "commit for question-9.File in lottery with <name>_1"
[master f5027e7] commit for question-9.File in lottery with <name>_1
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lottery_1.ipynb
create mode 100644 lottery_1.yml
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 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.
To https://github.com/CemBirbiri/vip_test.git
 fef47f5..f5027e7 master -> master
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

BRANCHING

16-

git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short"

17-

git config --list

18-

git hist

(The screen shot image is so long so I won't put it here)

```
vip_test — -zsh — 78x8
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git hist
* f5027e7 2021-10-04 | commit for question-9.File in lottery with <name>_1 (
HEAD -> my_branch, origin/master, master) [CemBirbiri]
* fef47f5 2021-10-04 | 2- commit for gitignore [CemBirbiri]
* 82d3592 2021-10-03 | commit for .gitignore (tag: addignore) [CemBirbiri]
* 67b87ee 2021-10-02 | cyclegan commit (tag: start_L.4) [CemBirbiri]
* 940b0a7 2021-10-02 | commit for vip test README [CemBirbiri]
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

19-

git checkout -b my_branch

git rev-parse --abbrev-ref HEAD (to check which branch I am working on)

20- Done

21-

touch file_from_my_branch

git add file_from_my_branch

git commit -m "question-21 file_from_my_branch"

git push

23-

git branch -a (to see all branches)

git checkout master

git status (to see which branch I am working on)

24-

git merge my_branch

25-

git branch -d my_branch

26-

git checkout -b new_branch

git rev-parse --abbrev-ref HEAD (to check which branch I am working on)

27-

echo "This line was added from new_branch" >> README.md

git add README.md

git commit -m "Commit from new_branch"

git push

28-

git checkout master

29-

echo "This line was added from master" >> README.md

git add README.md

git commit -m "Commit from master"

git push

30-

git merge new_branch

Auto-merging README.md

CONFLICT (content): Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

31- I should work on following text:

(1) <<<<<< HEAD

This line was added from master

(2) =====

This line was added from new_branch

(3) >>>>>> new_branch

I opened README.md file with “vim” and deleted the lines (1),(2),(3). Then:

git add README.md

git commit -m “after merge master and new_branch”

git push

32-

git hist

```
vip_test — -zsh — 78x16
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git hist
* 942bae1 2021-10-05 | after merge master and new_branch (HEAD -> master, or
igin/master) [CemBirbiri]
| \
| * 7878db5 2021-10-05 | Commit from new_branch (new_branch) [CemBirbiri]
| * 2e7d8ed 2021-10-05 | Commit from master [CemBirbiri]
| /
* a40cb32 2021-10-05 | question-21 file_from_my_branch [CemBirbiri]
* f5027e7 2021-10-04 | commit for question-9.File in lottery with <name>_1 [
CemBirbiri]
* fef47f5 2021-10-04 | 2- commit for gitignore [CemBirbiri]
* 82d3592 2021-10-03 | commit for .gitignore (tag: addignore) [CemBirbiri]
* 67b87ee 2021-10-02 | cyclegan commit (tag: start_L.4) [CemBirbiri]
* 940b0a7 2021-10-02 | commit for vip test README [CemBirbiri]
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
(base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

Old version:

33-

git checkout start_L.4

34-

git switch -c old_version

35-

git checkout master

36-

git status

...

On branch master

Your branch is up to date with 'origin/master'.

...

ls

README.md lottery_1.ipynb

file_from_my_branch lottery_1.yml

37-

git checkout start_L.4 environment.yml

error: pathspec 'environment.yml' did not match any file(s) known to git

38-

git status

ls

(There is no change)

39-

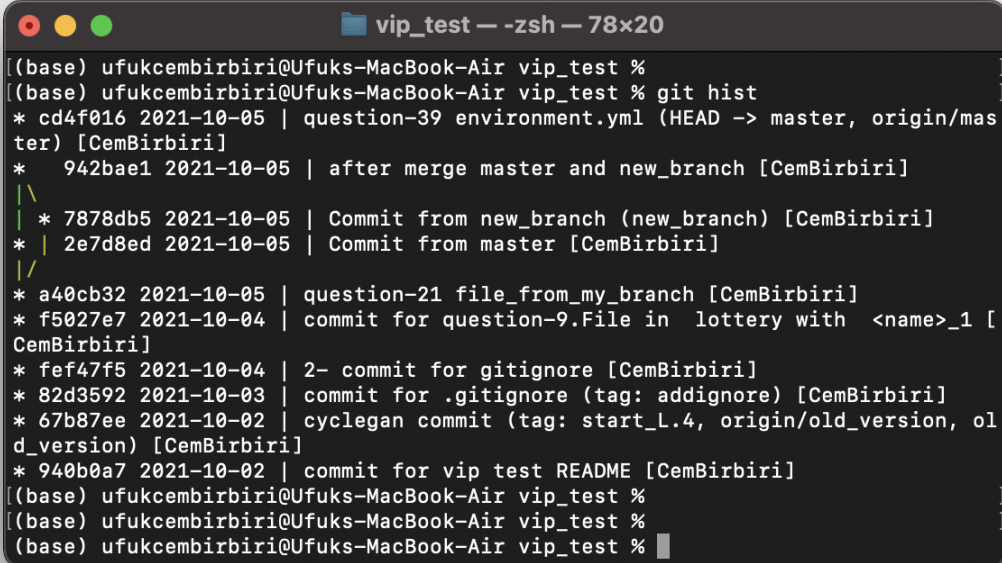
git add .

git commit -m "question-39 environment.yml"

git push

40-

git hist



```
vip_test --zsh-- 78x20
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test % git hist
* cd4f016 2021-10-05 | question-39 environment.yml (HEAD -> master, origin/master) [CemBirbiri]
* 942bae1 2021-10-05 | after merge master and new_branch [CemBirbiri]
| \
| * 7878db5 2021-10-05 | Commit from new_branch (new_branch) [CemBirbiri]
* | 2e7d8ed 2021-10-05 | Commit from master [CemBirbiri]
| /
* a40cb32 2021-10-05 | question-21 file_from_my_branch [CemBirbiri]
* f5027e7 2021-10-04 | commit for question-9.File in lottery with <name>_1 [CemBirbiri]
* fef47f5 2021-10-04 | 2- commit for gitignore [CemBirbiri]
* 82d3592 2021-10-03 | commit for .gitignore (tag: addignore) [CemBirbiri]
* 67b87ee 2021-10-02 | cyclegan commit (tag: start_L.4, origin/old_version, old_version) [CemBirbiri]
* 940b0a7 2021-10-02 | commit for vip test README [CemBirbiri]
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
((base) ufukcembirbiri@Ufuks-MacBook-Air vip_test %
```

41-

git reset --hard addignore

HEAD is now at 82d3592 commit for .gitignore

42-

git hist

43-

git hist --all

44-

git checkout 940b0a7

Note: switching to '940b0a7'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to false

HEAD is now at 940b0a7 commit for vip test README