esade

Version Control with Git

Do Good. Do Better.

Objectives

- Overview of version control
- How to create a Git repository
- How review changes in a Git repository
- How to make changes to a Git repository
- How to download files from a repository

esade

Introducing Version Control

Do Good. Do Better.

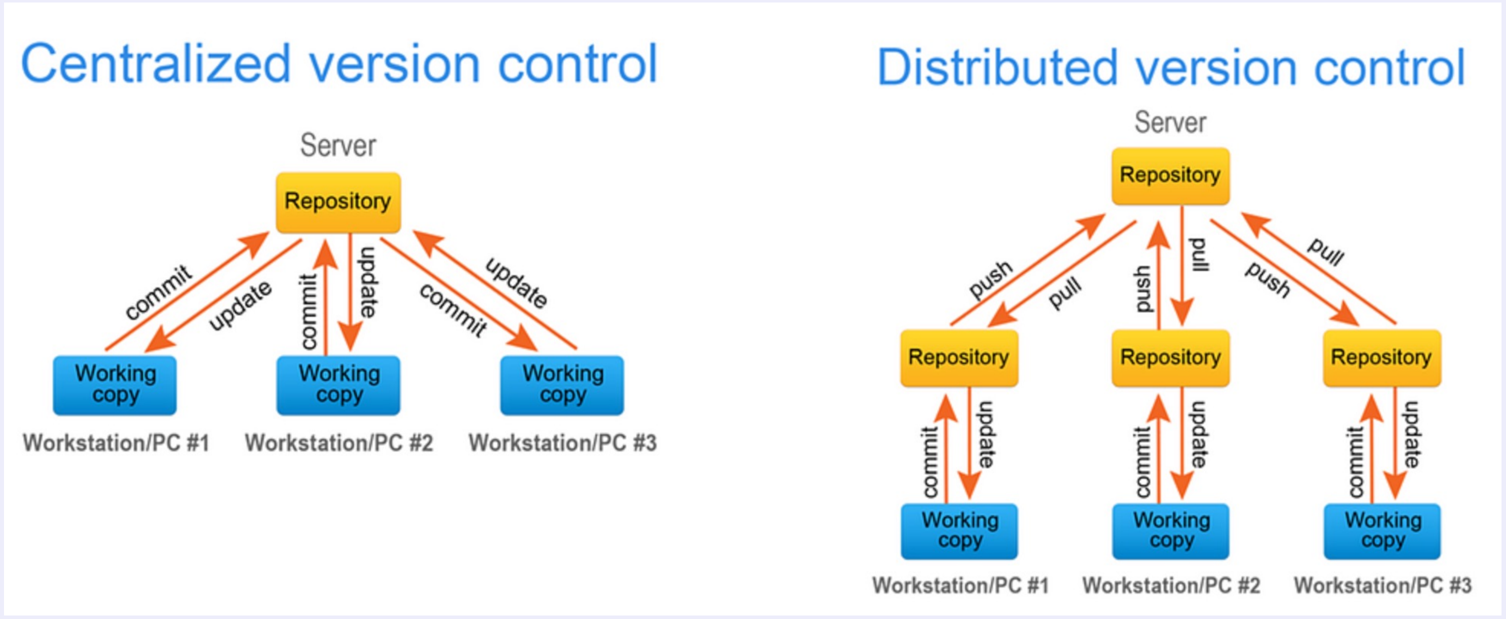
What is version control?

Helps to maintain a detailed history of a project

Provides the ability to work on different versions of the project

Provides the ability to return to any point in the project history to recover data or files

Centralized vs Distributed



Git vs GitHub

Git

Version control tool

GitHub

Service that hosts Git projects

A place you can upload your project

More about GitHub

GitHub is a code hosting platform for version control and collaboration

— Share your code
— Reuse someone else’s code
— Collaborate
— Create a profile

Terminology

— Commit: a save of your project’s files at a point in time

— Repository (repo): a directory containing your project

— Working directory: the directory in which you are working with your project files

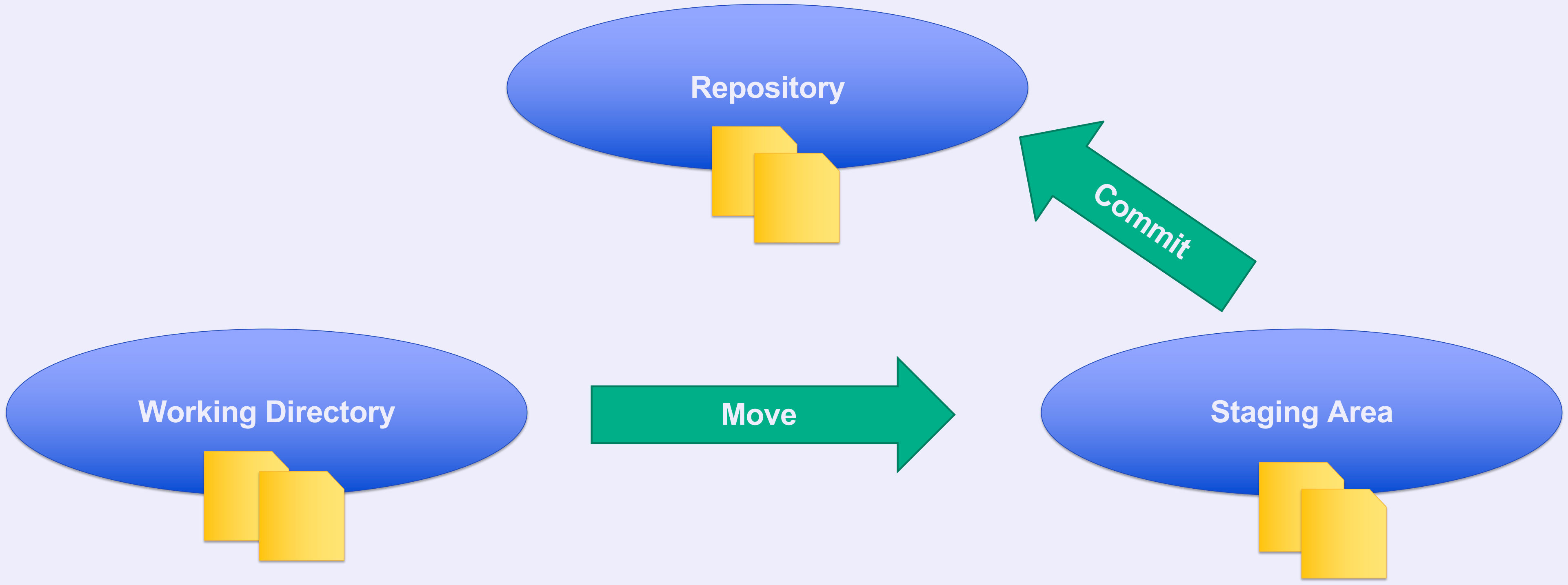
— Checkout: copying content from the repository to the Working directory

— Staging area: a file that stores information about the next commit

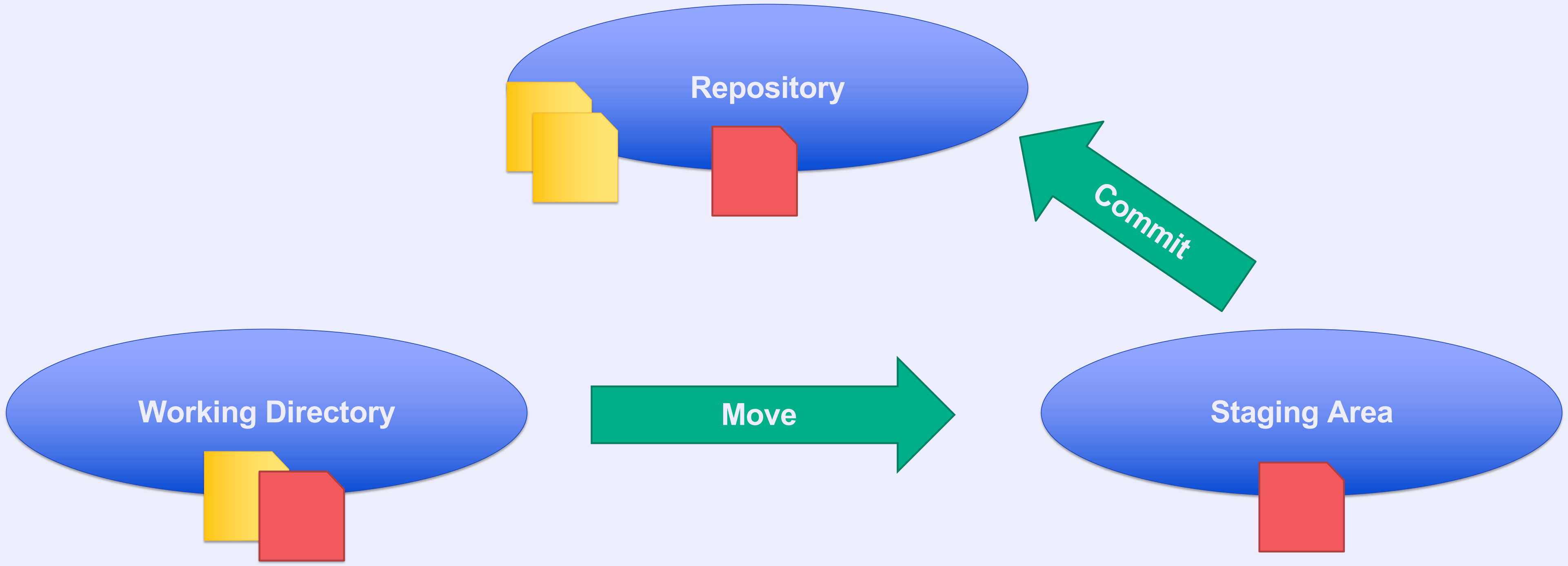
— SHA : an ID number for each commit

— Branch: it’s a new line of development when takes off from the main line (“a branch”).

Terminology – New



Terminology – Change



esade

Creating Git repository

Do Good. Do Better.

Git commands

— git clone: to copy a repository to your computer

Git config

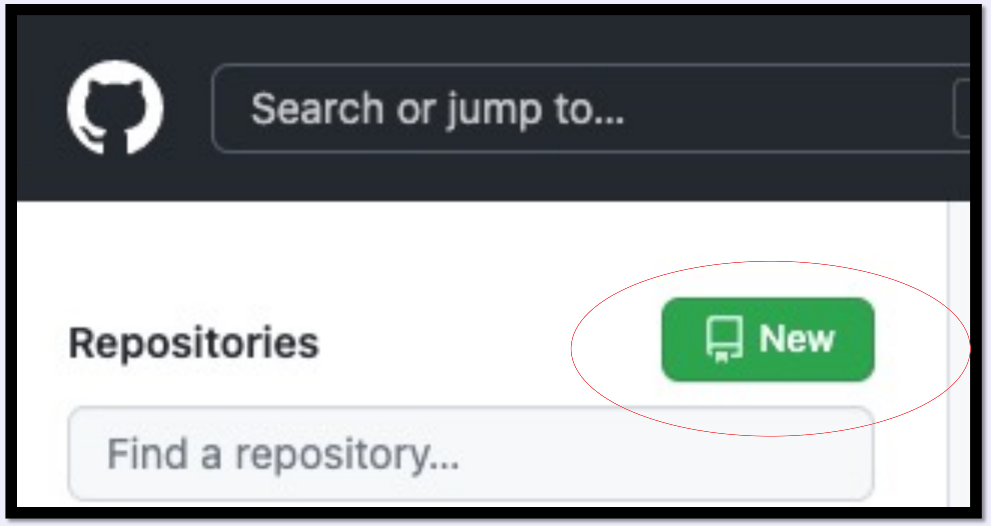
— git config - - global user.name “Mona Lisa”
— git config - - global user.email. “monalisa@email.com”

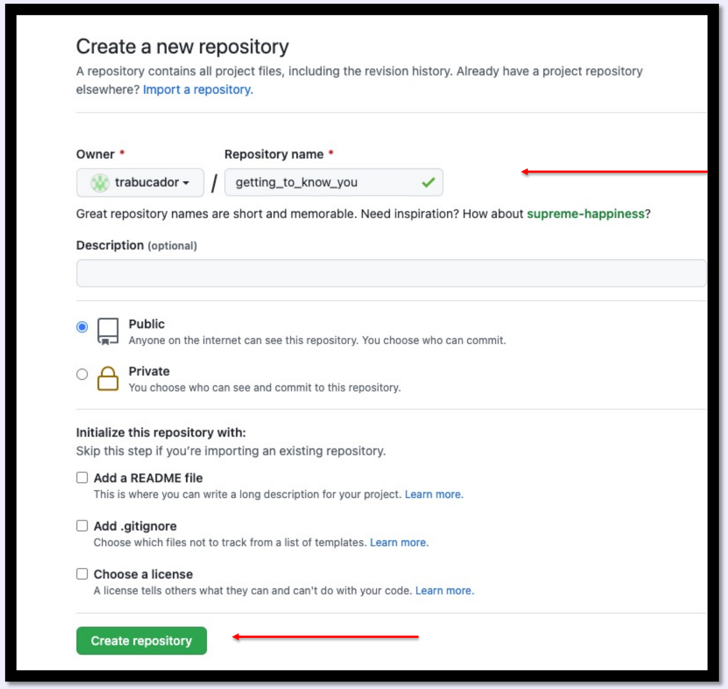
— Create personal access token -> this is your password
— https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token

— Only select repo

Create project repository

- Sign in to github.com




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * trabucador / Repository name * getting_to_know_you

Great repository names are short and memorable. Need inspiration? How about supreme-happiness?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. Learn more.

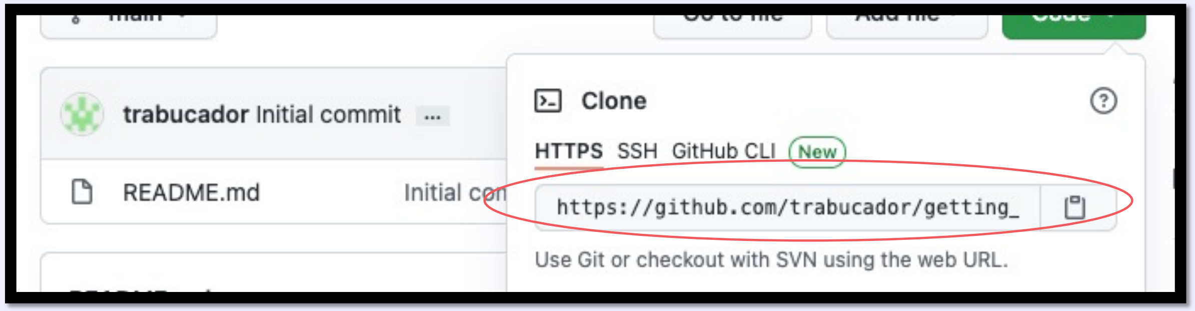
Add .gitignore
Choose which files not to track from a list of templates. Learn more.

Choose a license
A license tells others what they can and can't do with your code. Learn more.

Create repository

Clone repository

git clone url_of_repository



trabucador Initial commit

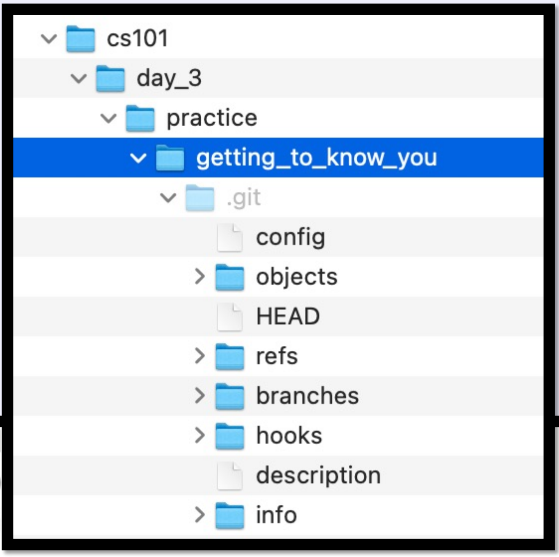
README.md Initial com

Clone

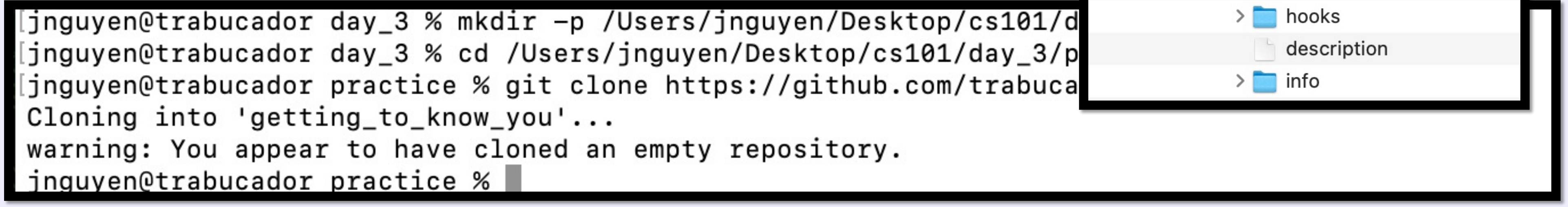
HTTPS SSH GitHub CLI New

```
https://github.com/trabucador/getting_
```

Use Git or checkout with SVN using the web URL.



```
cs101
  day_3
    practice
      getting_to_know_you
        .git
          config
          objects
          HEAD
          refs
          branches
          hooks
          description
          info
```



```
jnguyen@trabucador day_3 % mkdir -p /Users/jnguyen/Desktop/cs101/d
jnguyen@trabucador day_3 % cd /Users/jnguyen/Desktop/cs101/day_3/p
jnguyen@trabucador practice % git clone https://github.com/trabuca
Cloning into 'getting_to_know_you'...
warning: You appear to have cloned an empty repository.
jnguyen@trabucador practice %
```

esade

Saving changes to repo

Do Good. Do Better.

Git commands

— git add: add files from the working directory to the staging area

— git status: to determine status of a repository

— git commit: saves the files in the staging area to the repository

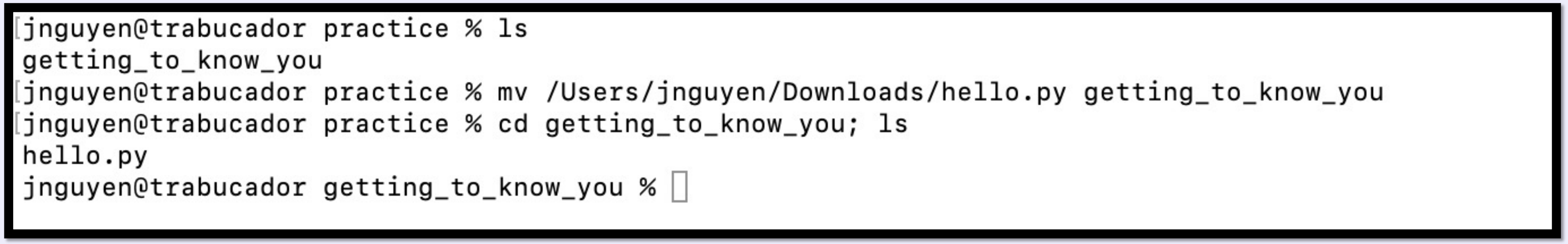
Push notebook to GitHub repository

1) Move notebook to repository
2) Start tracking notebook
3) Check that notebook is being tracked
4) Create a checkpoint that we can revert back to

Push notebook to repository

Move notebook to repository *mv path_of_nb path_of_proj_dir*

- Go to CLI
- Move downloaded file from downloads directory to project directory



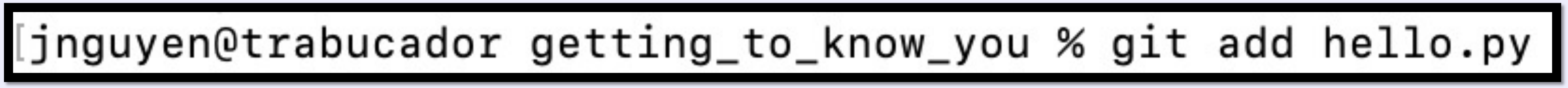
```
jnguyen@trabucador practice % ls
getting_to_know_you
jnguyen@trabucador practice % mv /Users/jnguyen/Downloads/hello.py getting_to_know_you
jnguyen@trabucador practice % cd getting_to_know_you; ls
hello.py
jnguyen@trabucador getting_to_know_you % 
```

Push notebook to repository

Start tracking notebook, move to staging

git add file_name

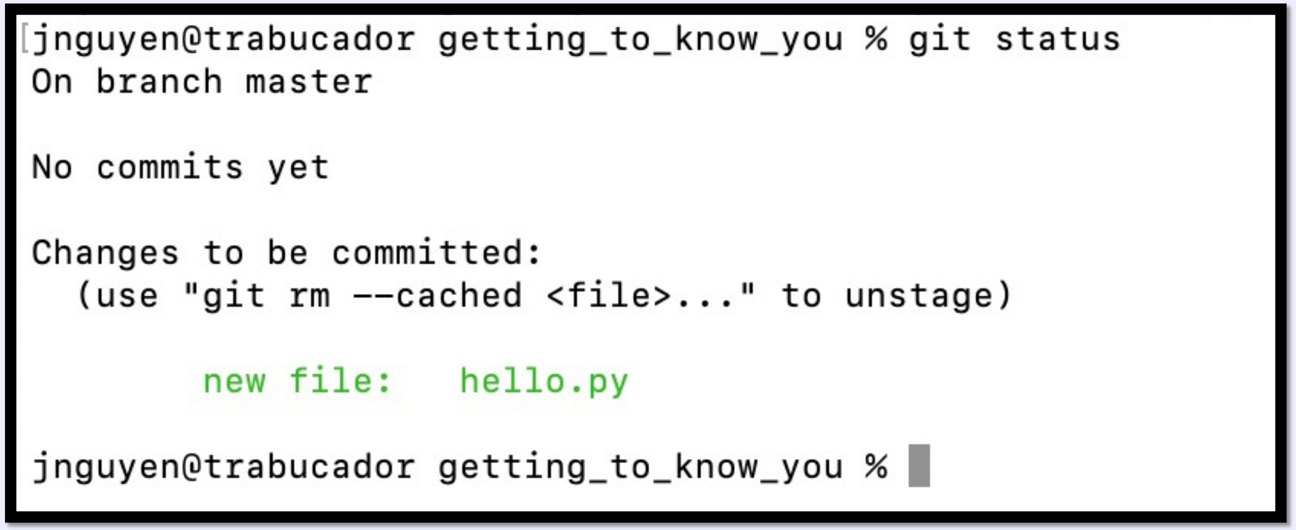
- Go to project directory (getting_to_know_you)
- Tell git to start tracking our notebook (git add)



```
[jnguyen@trabucador getting_to_know_you % git add hello.py
```

Saving changes to repo

Push notebook to repository (check status)
git status



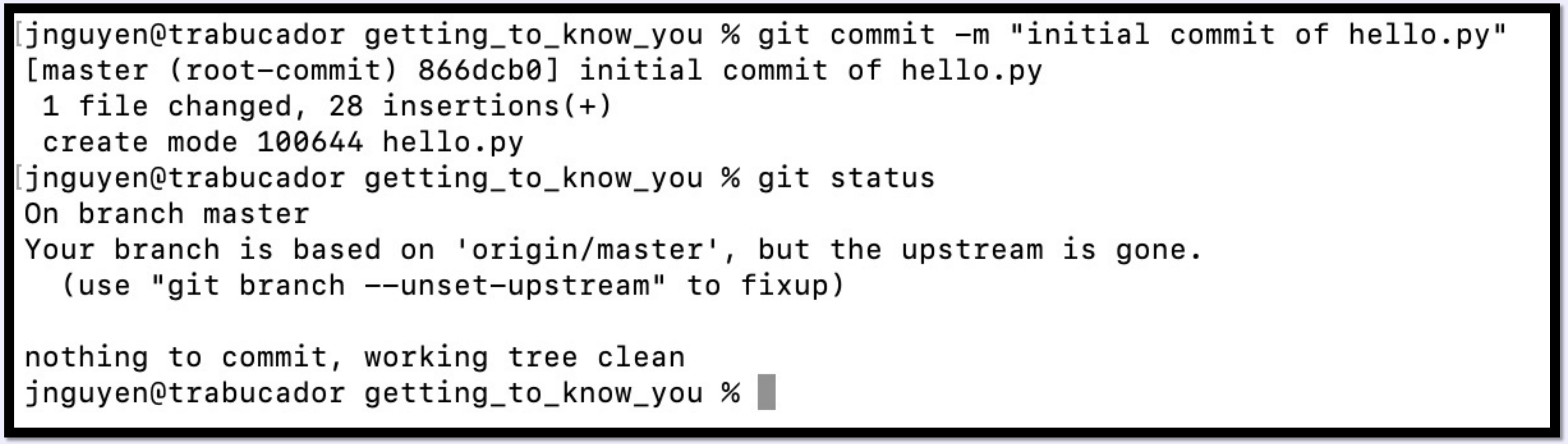
```
[jnguyen@trabucador getting_to_know_you % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   hello.py

jnguyen@trabucador getting_to_know_you % 
```

esade

Saving changes to repo

Push notebook to repository (create a version to go back to)

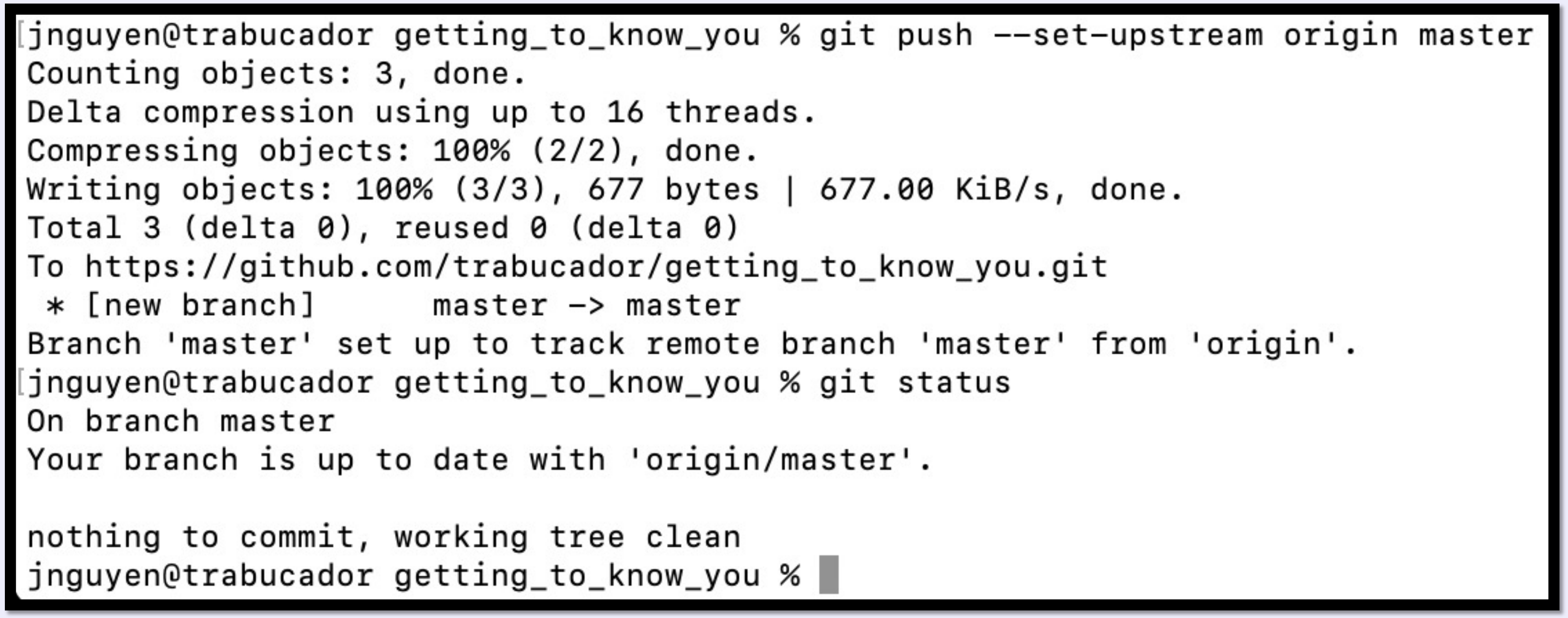
git commit –m note_of_changes_made



```
[jnguyen@trabucador getting_to_know_you % git commit -m "initial commit of hello.py"
[master (root-commit) 866dcb0] initial commit of hello.py
 1 file changed, 28 insertions(+)
 create mode 100644 hello.py
[jnguyen@trabucador getting_to_know_you % git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
jnguyen@trabucador getting_to_know_you % 
```

Push notebook to repository (push version to GitHub)

git push

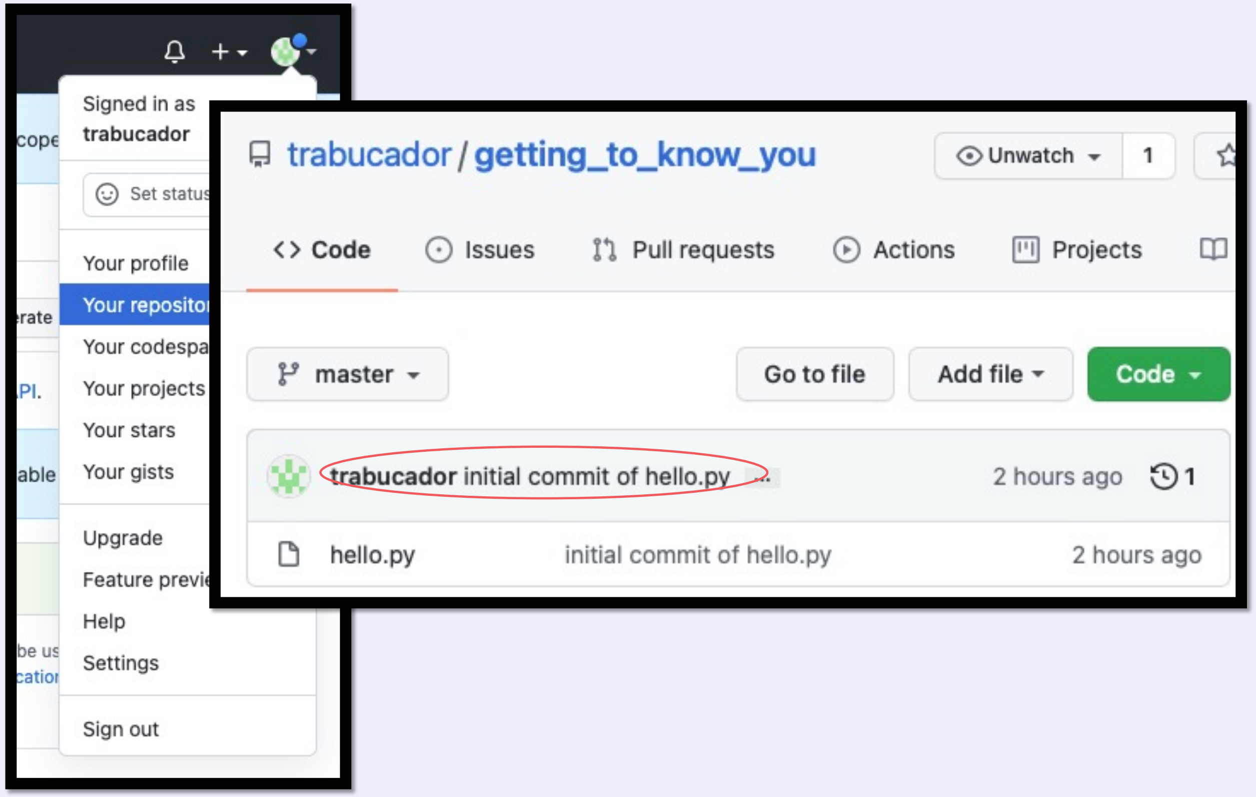


```
jnguyen@trabucador getting_to_know_you % git push --set-upstream origin master
Counting objects: 3, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 677 bytes | 677.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/trabucador/getting_to_know_you.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
jnguyen@trabucador getting_to_know_you % git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
jnguyen@trabucador getting_to_know_you % 
```

esade

Push notebook to repository (view changes)

- Go to github.com
- Select getting_to_know_you
- Note comment you made
- Select file to view



trabucador initial commit of hello.py ... 2 hours ago 1

hello.py initial commit of hello.py 2 hours ago

esade

Branching

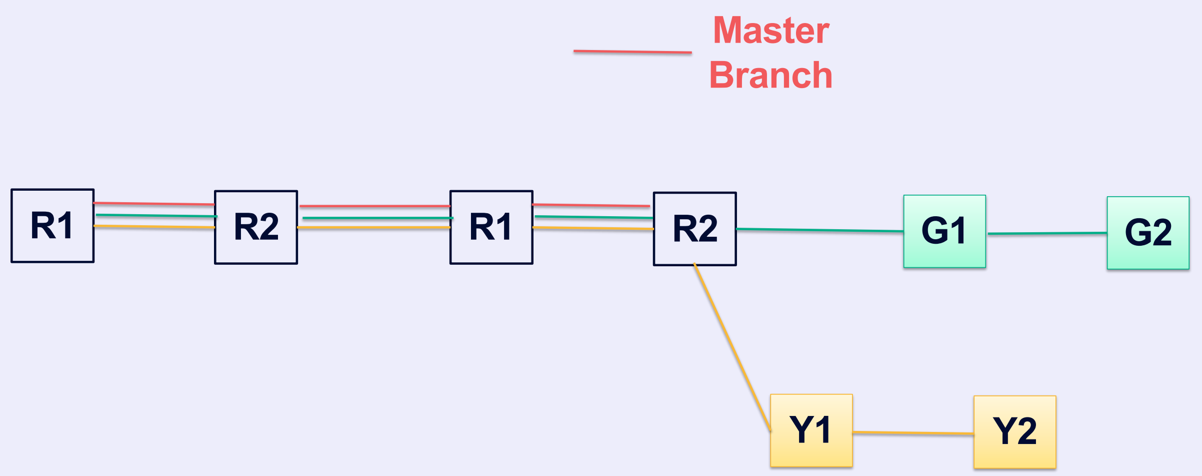
Do Good. Do Better.

Git commands

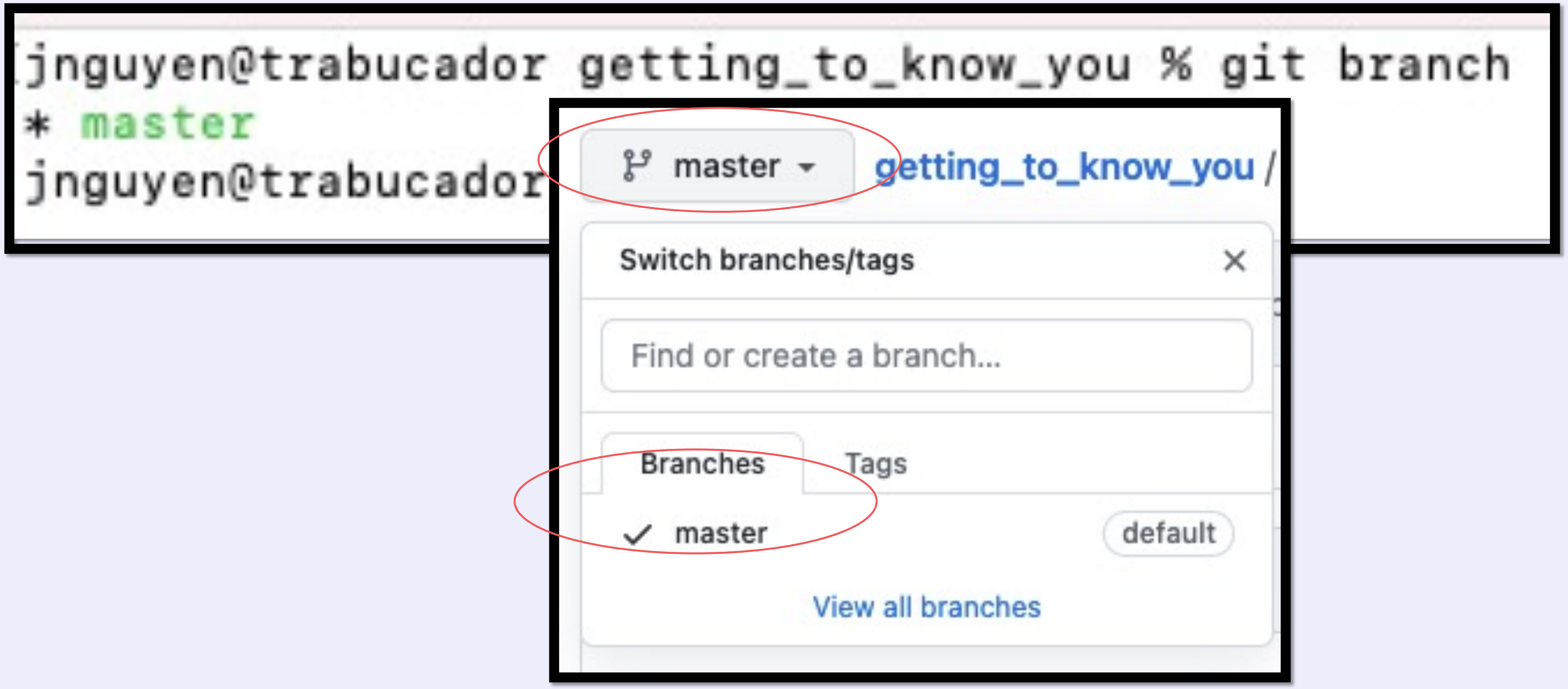
— git branch: allows you to create different features of your project in parallel

— git checkout: allows you to switch between different branches

Branching

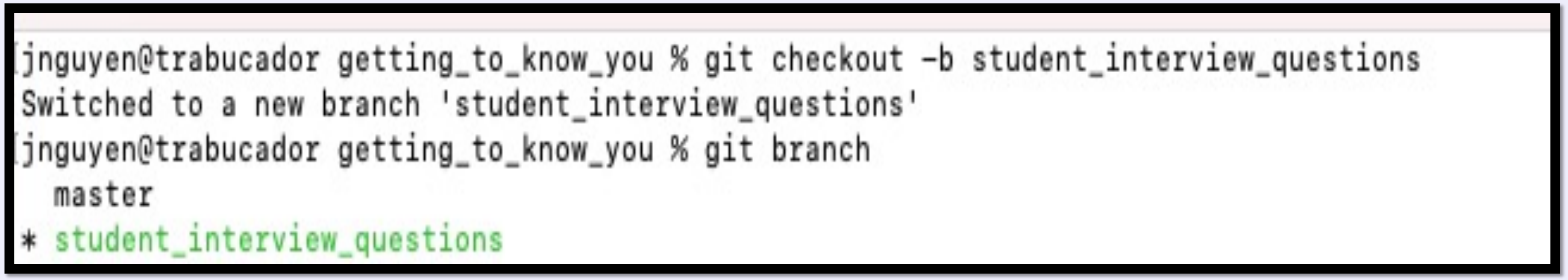


List the branches in our project
git branch



```
jnguyen@trabucador getting_to_know_you % git branch
* master
jnguyen@trabucador
```

Create and change to new branch git checkout –b *branch_name*

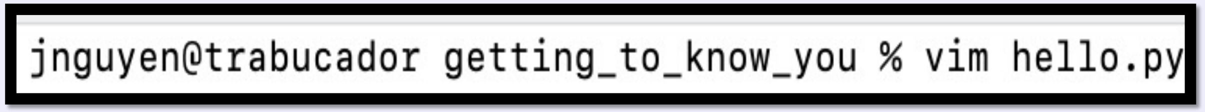


```
[jnguyen@trabucador getting_to_know_you % git checkout -b student_interview_questions
Switched to a new branch 'student_interview_questions'
[jnguyen@trabucador getting_to_know_you % git branch
  master
* student_interview_questions
```

How do we know which branch we are working on?
A commit made now, will affect which branch?

Making changes to hello.py on new branch (student_interview_questions)

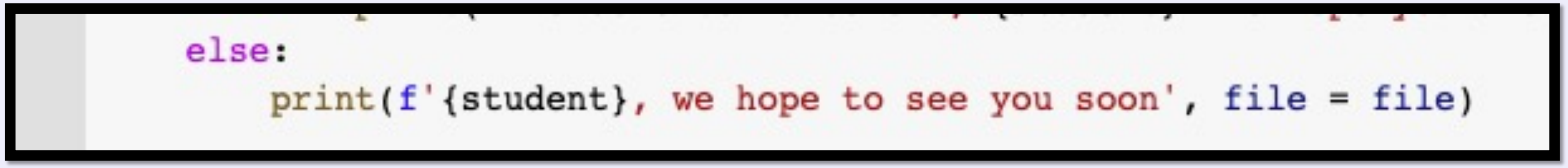
- Open vim text editor



```
jnguyen@trabucador getting_to_know_you % vim hello.py
```

- Vim commands

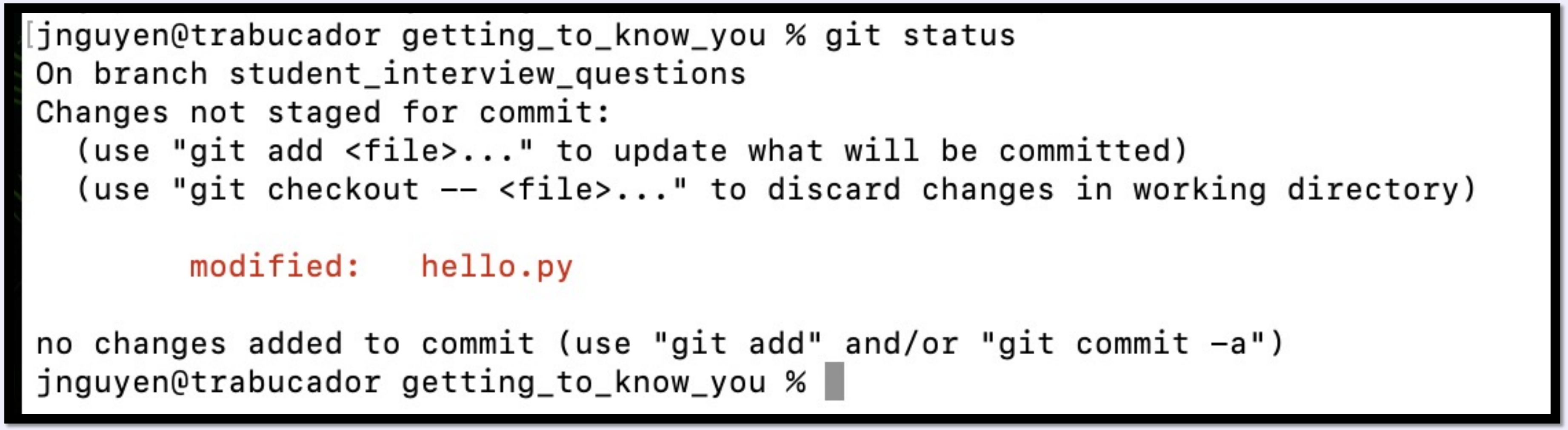
- " i " to insert message
- " esc " to exit inset mode
- " : " to enter command mode and save changes
- "wq" to write and quit the message



```
else:
    print(f'{student}, we hope to see you soon', file = file)
```

Making changes to hello.py on new branch (student_interview_questions)

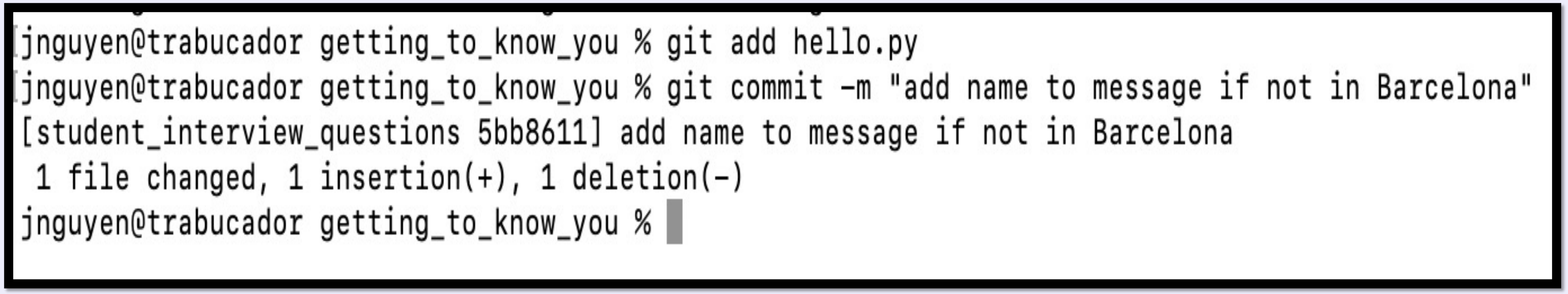
- Check to see that git recognizes file has been modified



```
jnguyen@trabucador getting_to_know_you % git status
On branch student_interview_questions
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   hello.py

no changes added to commit (use "git add" and/or "git commit -a")
jnguyen@trabucador getting_to_know_you %
```

Making changes to hello.py on new branch (student_interview_questions)

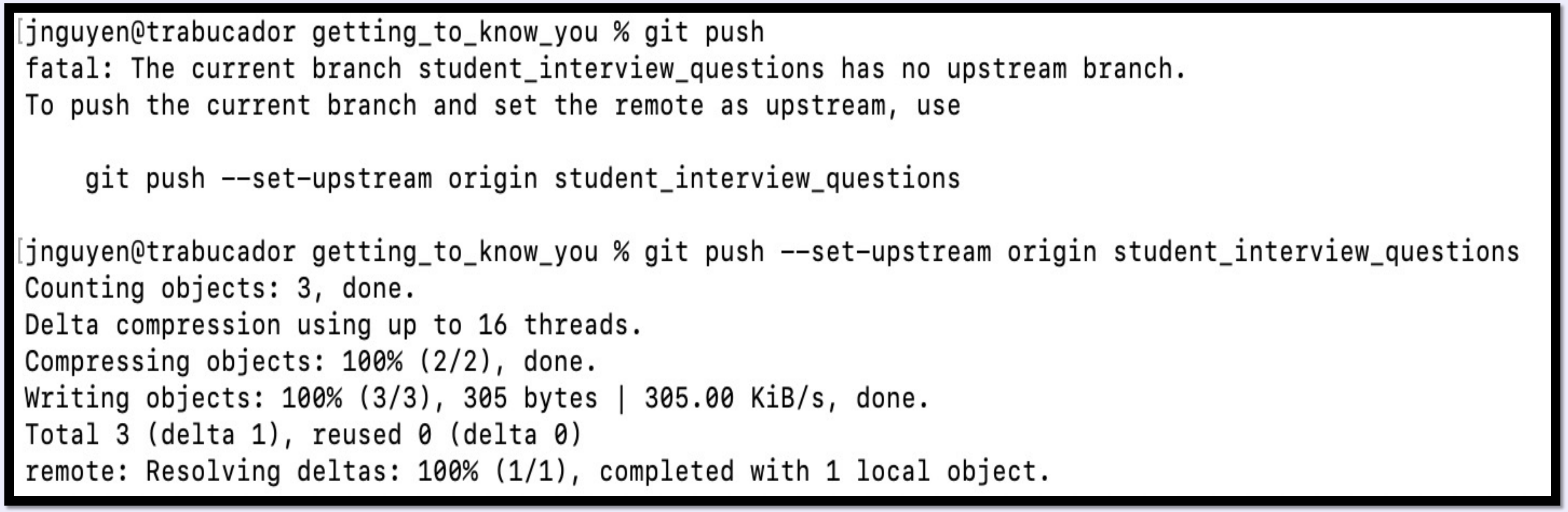
- git add (move files to staging area)
- git commit



```
jnguyen@trabucador getting_to_know_you % git add hello.py
jnguyen@trabucador getting_to_know_you % git commit -m "add name to message if not in Barcelona"
[student_interview_questions 5bb8611] add name to message if not in Barcelona
 1 file changed, 1 insertion(+), 1 deletion(-)
jnguyen@trabucador getting_to_know_you %
```

Making changes to hello.py on new branch (student_interview_questions)

- git push



```
[jnguyen@trabucador getting_to_know_you % git push
fatal: The current branch student_interview_questions has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin student_interview_questions

[jnguyen@trabucador getting_to_know_you % git push --set-upstream origin student_interview_questions
Counting objects: 3, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```

esade

Review version history

Do Good. Do Better.

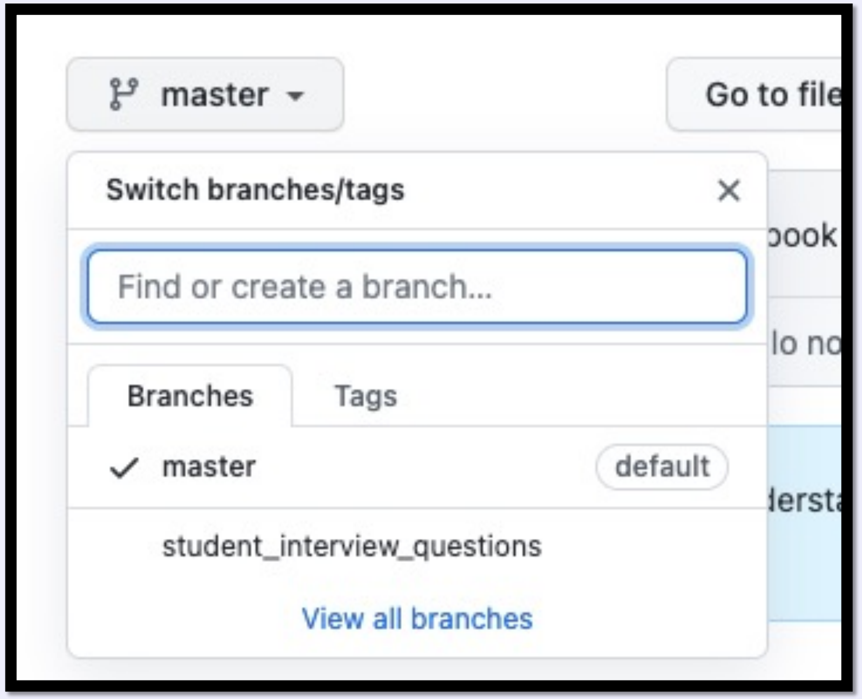
Git commands

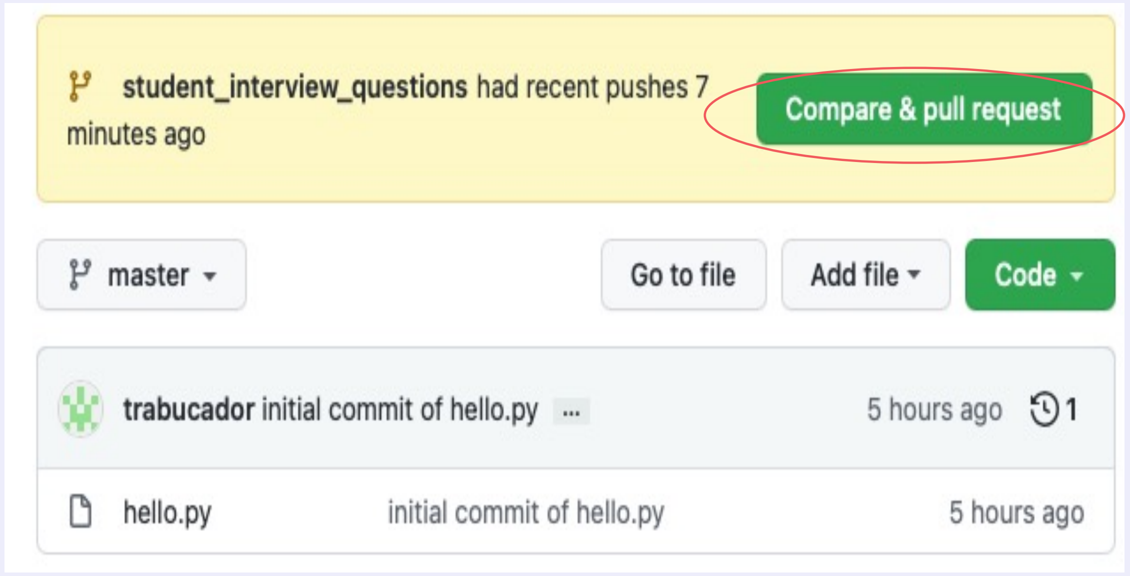
— git log: get information about existing commits

— git show: displays information about a specific commit

Reviewing commit history

- From the GitHub repository



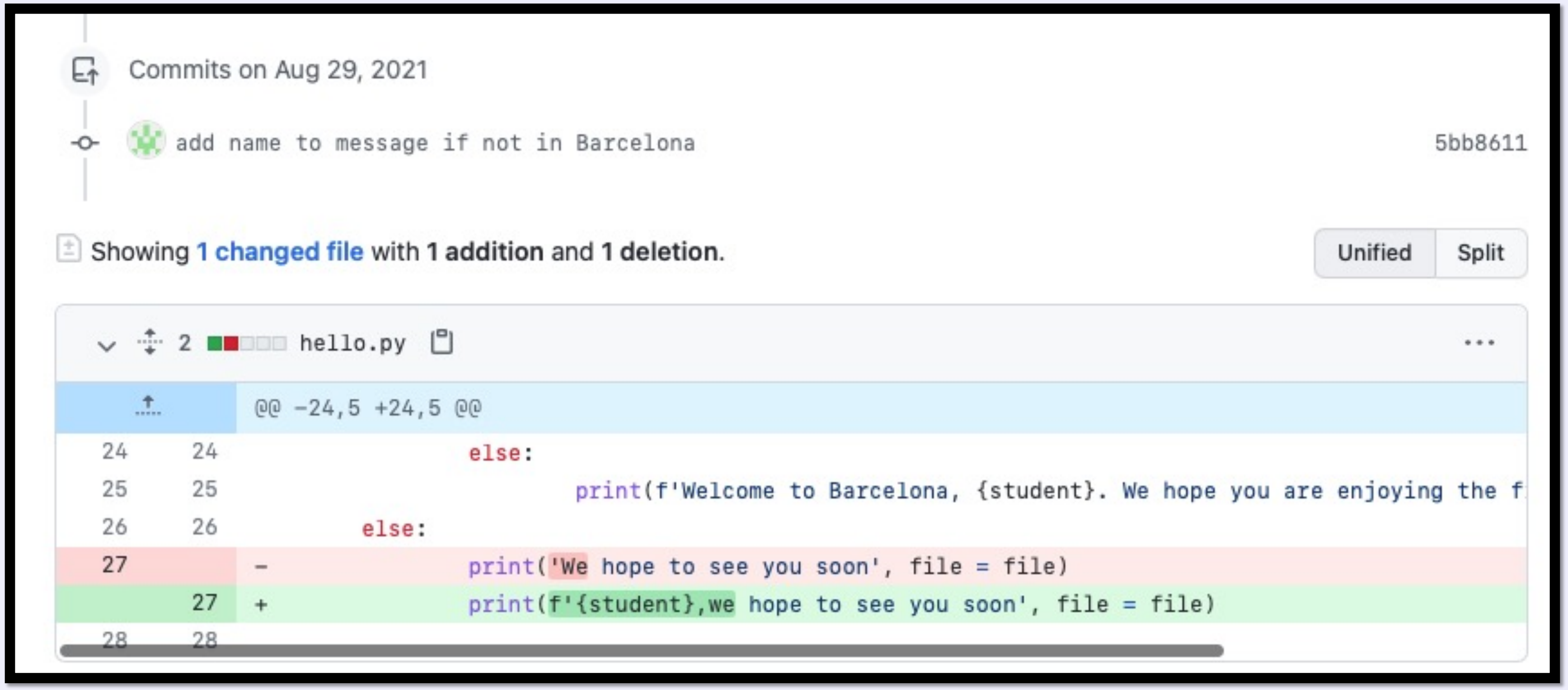

student_interview_questions had recent pushes 7 minutes ago

trabucador initial commit of hello.py ... 5 hours ago 1

hello.py initial commit of hello.py 5 hours ago

Reviewing commit history

- From the GitHub repository



Commits on Aug 29, 2021

add name to message if not in Barcelona 5bb8611

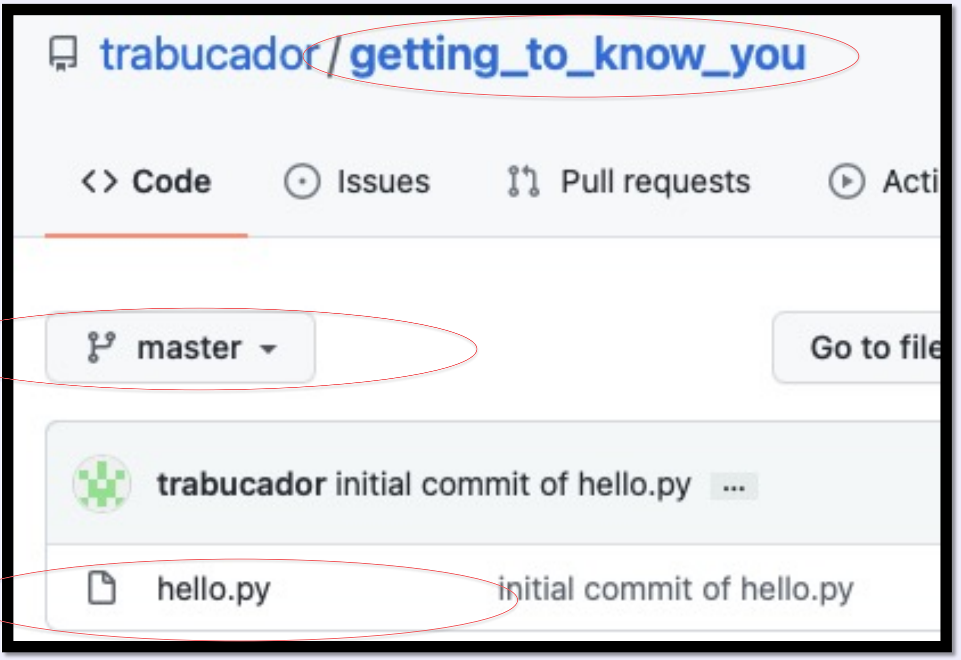
Showing 1 changed file with 1 addition and 1 deletion.

2 hello.py

```
@@ -24,5 +24,5 @@
24  24                  else:
25  25                          print(f'Welcome to Barcelona, {student}. We hope you are enjoying the f
26  26          else:
27      -               print('We hope to see you soon', file = file)
    27  +               print(f'{student},we hope to see you soon', file = file)
28  28
```

Reviewing commit history

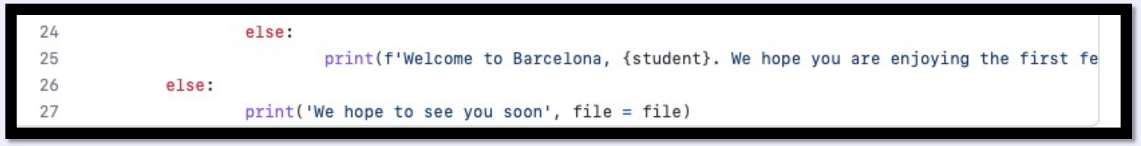
- We can check to see that no changes were made to the master



trabucador / getting_to_know_you

trabucador initial commit of hello.py ...

hello.py initial commit of hello.py



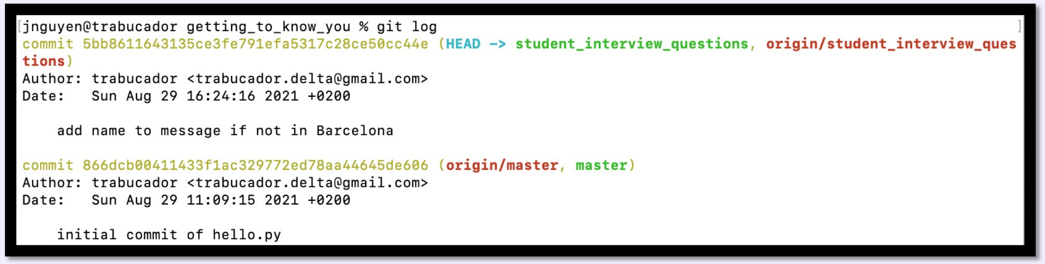
```
		else:
			print(f'Welcome to Barcelona, {student}. We hope you are enjoying the first fe
	else:
		print('We hope to see you soon', file = file)
```

Reviewing commit history

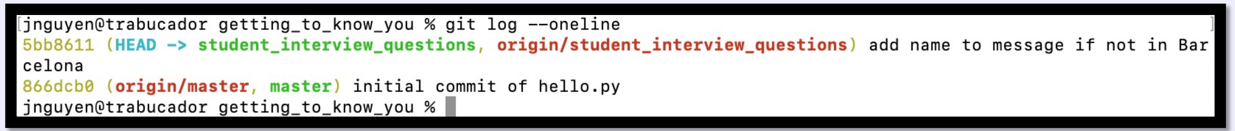
git log from project dir

git log --oneline from project dir

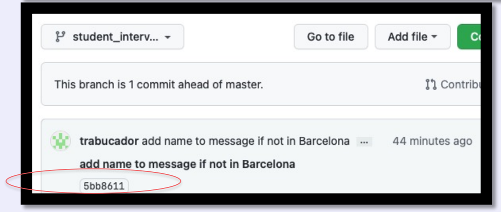
- Commit SHA
- Who made the commit
- Date of the commit
- Description of the commit
- : indicates that there are more commits than shown



```
jnguyen@trabucador getting_to_know_you % git log
commit 5bb8611643135ce3fe791efa5317c28ce50cc44e (HEAD -> student_interview_questions, origin/student_interview_ques
tions)
Author: trabucador <trabucador.delta@gmail.com>
Date:   Sun Aug 29 16:24:16 2021 +0200

    add name to message if not in Barcelona

commit 866dcb00411433f1ac329772ed78aa44645de606 (origin/master, master)
Author: trabucador <trabucador.delta@gmail.com>
Date:   Sun Aug 29 11:09:15 2021 +0200

    initial commit of hello.py
```



```
jnguyen@trabucador getting_to_know_you % git log --oneline
5bb8611 (HEAD -> student_interview_questions, origin/student_interview_questions) add name to message if not in Bar
celona
866dcb0 (origin/master, master) initial commit of hello.py
jnguyen@trabucador getting_to_know_you %
```



This branch is 1 commit ahead of master.

trabucador add name to message if not in Barcelona ... 44 minutes ago

add name to message if not in Barcelona

5bb8611

esade

Exercise (group)

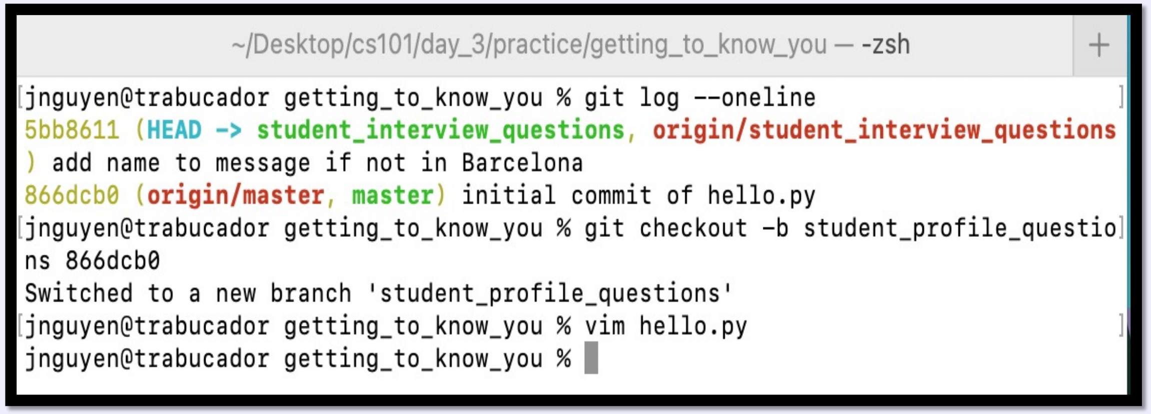
- Let’s assume we don’t like the changes
- Let’s create a branch that takes us back to before our change
- To do that we have to create a branch

- Find the SHA of the commit prior to the change
- In this example its 866dcb0

- Create a branch from this point to build on
- Call the new branch student_profile_questions
- Switch to it

- Open hello.py in the new branch to see that it’s the previous version

Note we are in the directory of the project repository



```
jnguyen@trabucador getting_to_know_you % git log --oneline
5bb8611 (HEAD -> student_interview_questions, origin/student_interview_questions
) add name to message if not in Barcelona
866dcb0 (origin/master, master) initial commit of hello.py
jnguyen@trabucador getting_to_know_you % git checkout -b student_profile_questio
ns 866dcb0
Switched to a new branch 'student_profile_questions'
jnguyen@trabucador getting_to_know_you % vim hello.py
jnguyen@trabucador getting_to_know_you %
```

Exercise (group)

- Create a notebook called student_questions.ipynb
- Follow the example notebook and create questions to get to know the students in the class
- Make sure that you can execute it
- Save the file

- Add this notebook to your getting_to_know_you repository
- Push it to GitHub
- Share the link with another group

- Download another group's student_questions.ipynb notebook
- Execute it in colab for each member of the group
- Save your group's notebook. Make sure that all members' full names are listed in the notebook
- Upload the file to the moodle

esade

Do Good. Do Better.