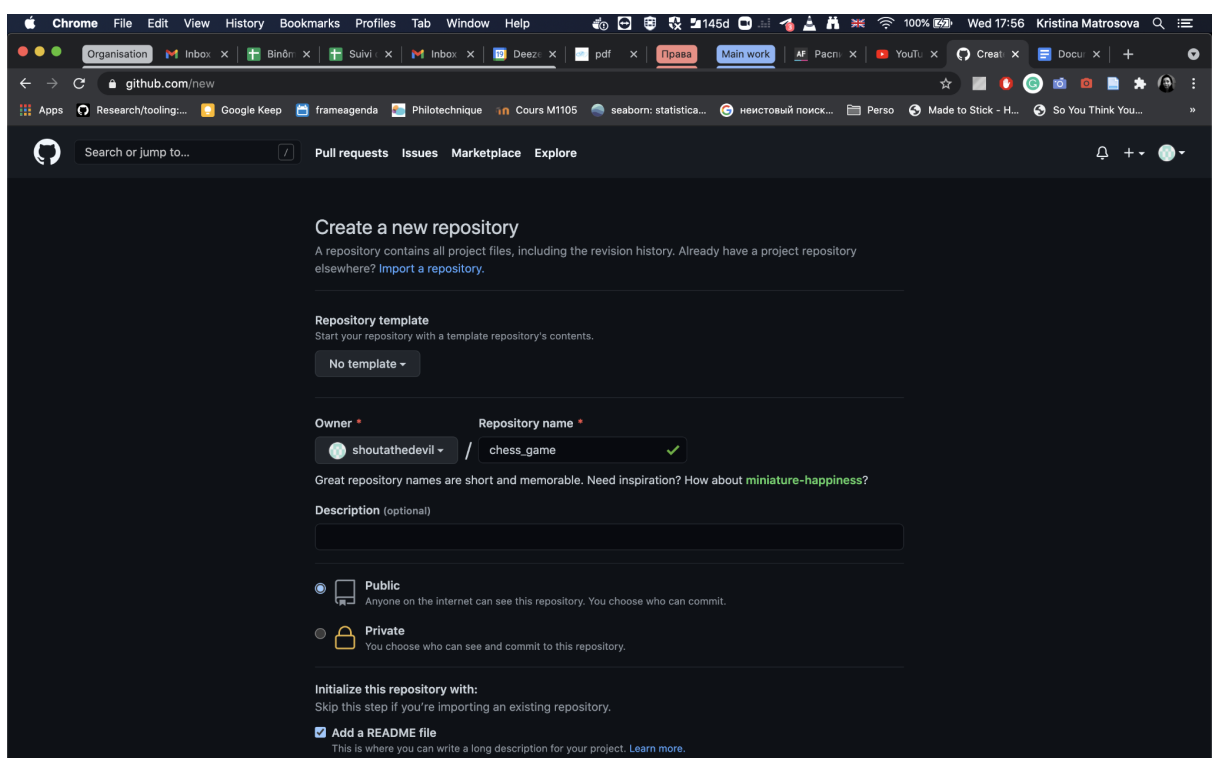
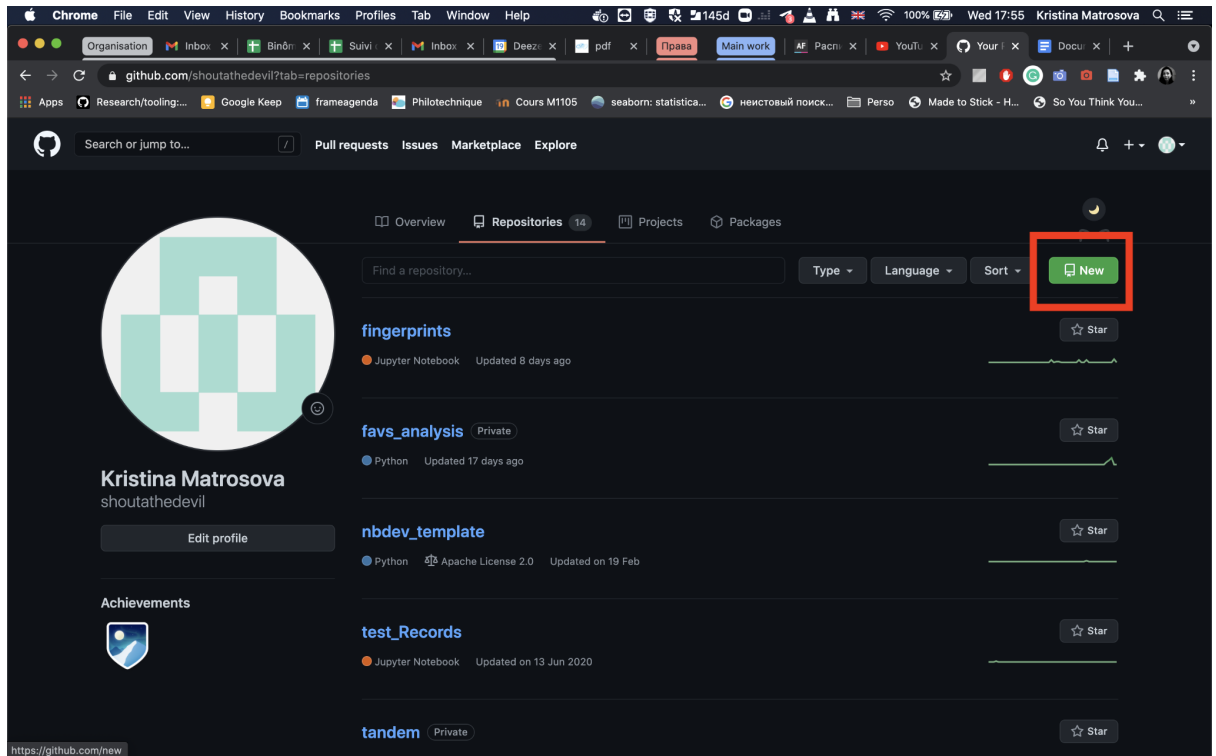


# Tuto Github

1. Créer un compte perso
2. Créer un repo (un seul par binôme)

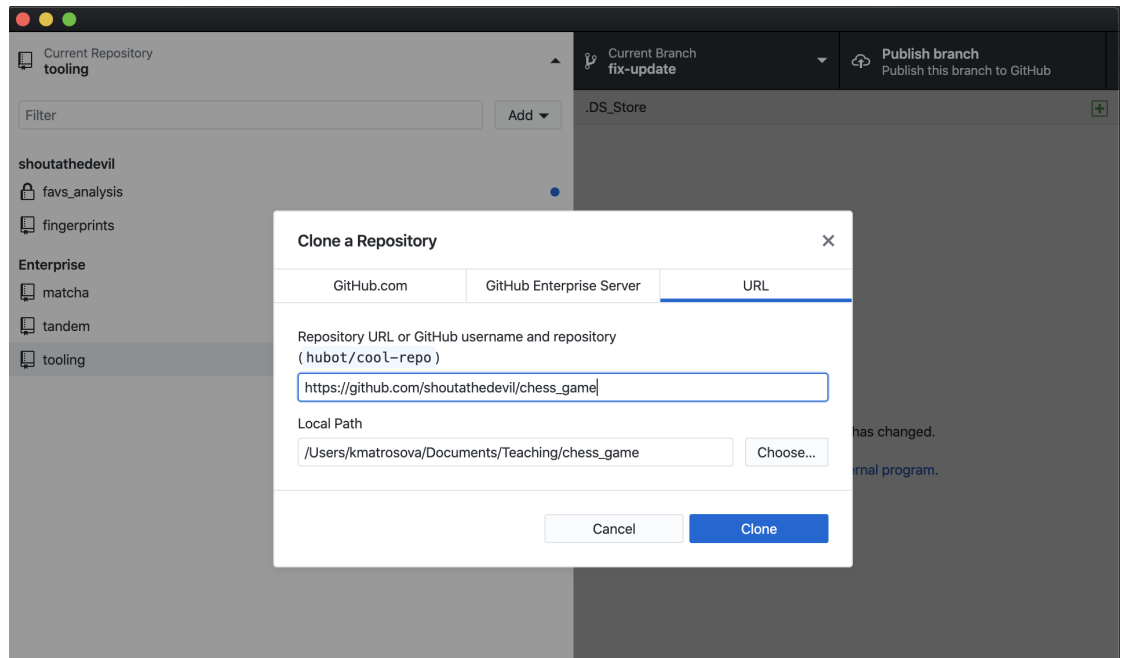


3. Cloner le repo sur sa machine (pour chaque binôme)
  - a. Avec la commande **git clone** dans le terminal

```
Teaching — -bash — 90x19

FRLPMC0764:Teaching kmatrosova$ git clone https://github.com/shoutathedevil/chess_game
Cloning into 'chess_game'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
FRLPMC0764:Teaching kmatrosova$
```

- b. Vous pouvez aussi utiliser GitHub Desktop sur Mac et Windows (lien de téléchargement <https://desktop.github.com/>)



4. Pour récupérer la dernière version du projet, utilisez la commande **git pull**. Utilisez cette commande systématiquement avant toute modification du code: il est possible que votre binôme ait avancé sur votre projet sans que vous soyez au courant et donc le code sur votre machine n'est plus à jour. Assurez-vous d'être dans le bon repo lorsque vous entrez cette commande.

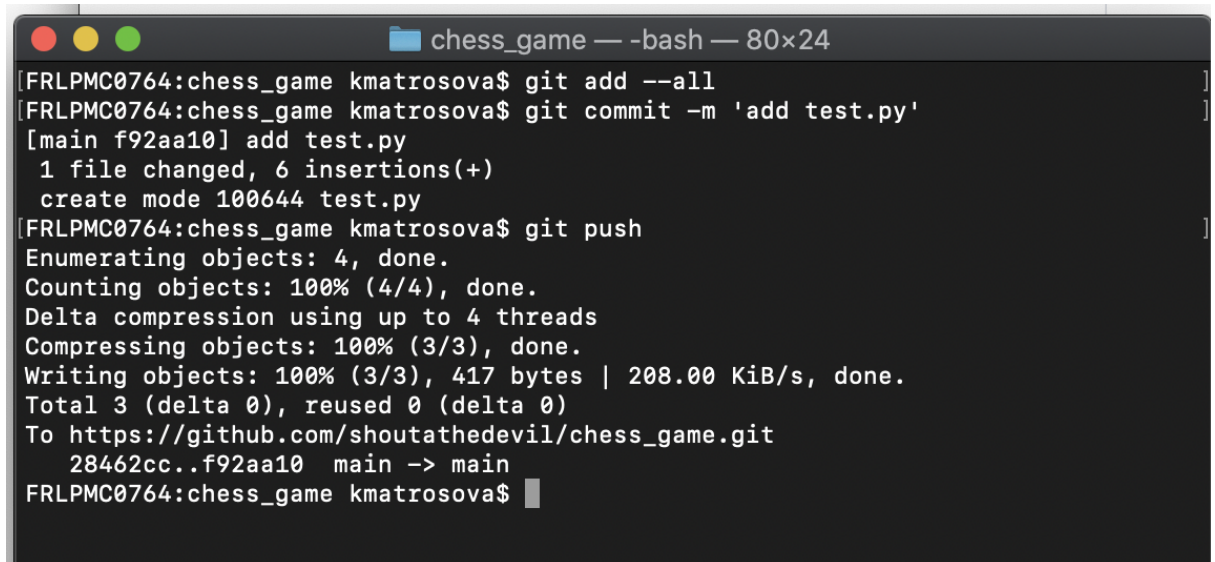
```
chess_game — -bash — 90x19

FRLPMC0764:Teaching kmatrosova$ cd chess_game/
FRLPMC0764:chess_game kmatrosova$ git pull
Already up to date.
FRLPMC0764:chess_game kmatrosova$
```

5. Une fois que vous avez fini vos modifications, vous voulez les joindre au code principal.
- Si vous avez créé de nouveaux fichiers, utilisez la commande **git add**. Vous pouvez utiliser **git add --all** si vous avez créé plusieurs nouveaux fichiers et que vous voulez tous les ajouter en même temps.
  - Maintenant il faut faire un *commit*, c'est-à-dire mettre ensemble tous les fichiers modifiés / créés et les labelliser avec un message. Attention à mettre

des messages représentatifs des modifications faites, pour faciliter la vie à vos collaborateurs (et parfois à vous même) lorsqu'ils voudront regarder ce que vous avez modifié dans le code. Pour cela vous utiliserez **git commit -m "mon message"**.

- c. Finalement, on utilise la commande **git push** pour charger toutes les modifications dans le repo principal.

A terminal window titled 'chess\_game — -bash — 80x24' with a dark background and light text. It shows the execution of git commands: 'git add --all', 'git commit -m 'add test.py'', and 'git push'. The output includes commit details like '[main f92aa10] add test.py', file changes, and the push progress to a GitHub repository.

```
FRLPMC0764:chess_game kmatrosova$ git add --all
FRLPMC0764:chess_game kmatrosova$ git commit -m 'add test.py'
[main f92aa10] add test.py
 1 file changed, 6 insertions(+)
 create mode 100644 test.py
FRLPMC0764:chess_game kmatrosova$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 417 bytes | 208.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/shoutathedevil/chess_game.git
 28462cc..f92aa10  main -> main
FRLPMC0764:chess_game kmatrosova$
```