



Versionning

1.0

CYRILLE FRANÇOIS © ATENO TECH

3

.3

Les meilleurs tutoriels Git

- <http://pcottle.github.io/learnGitBranching/>
- <https://try.github.io/levels/1/challenges/1>¹

Suivre ce tutoriel pour découvrir simplement les interactions entre vos commandes Git et les fichiers dans votre environnement de développement :

Suivre ce tutoriel pour découvrir la puissance de Git lors de la gestion de vos branches :

- <http://pcottle.github.io/learnGitBranching/>³

Workshop pour découvrir Git avec Github

- Une rapide présentation : Git & EGit beginner Workshop
- Un ensemble d'exercices pratiques : Repository du workshop

Créer un compte github

Lier un projet local à un git distant :

1. Initialiser le git local
2. Lier le git local au distant
3. Modifier le local et commiter les modifications
4. Pousser le git local sur le distant

1 - <https://try.github.io/levels/1/challenges/1>

2 - <https://try.github.io/levels/1/challenges/1>

3 - <http://pcottle.github.io/learnGitBranching/>

C'est prêt :)

Indices :

Initialiser le git local :

- Se placer dans le répertoire projet
- Initialiser le git local :

```
git init
```

Lier le git local au distant à l'URL <git@ADRESSE_DU_GIT_DISTANT> :

```
git remote add origin <git@ADRESSE_DU_GIT_DISTANT>
```

Modifier le local et commiter les modifications :

```
git add .
```

```
git commit -m "Init"
```

Vérifier que tout est bien dans le git local

```
D:\Dev\AtenoTech\workshopgit\atelier-git>git status
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   .gitignore
new file:   README.md
new file:   WebContent/index.html
new file:   WebContent/profils/Cyrille.html
new file:   WebContent/profils/Test.html
```

Init du git local : git status

Pousser le git local sur le distant :

```
git push -u origin master
```

Question 2

Importer le projet Git

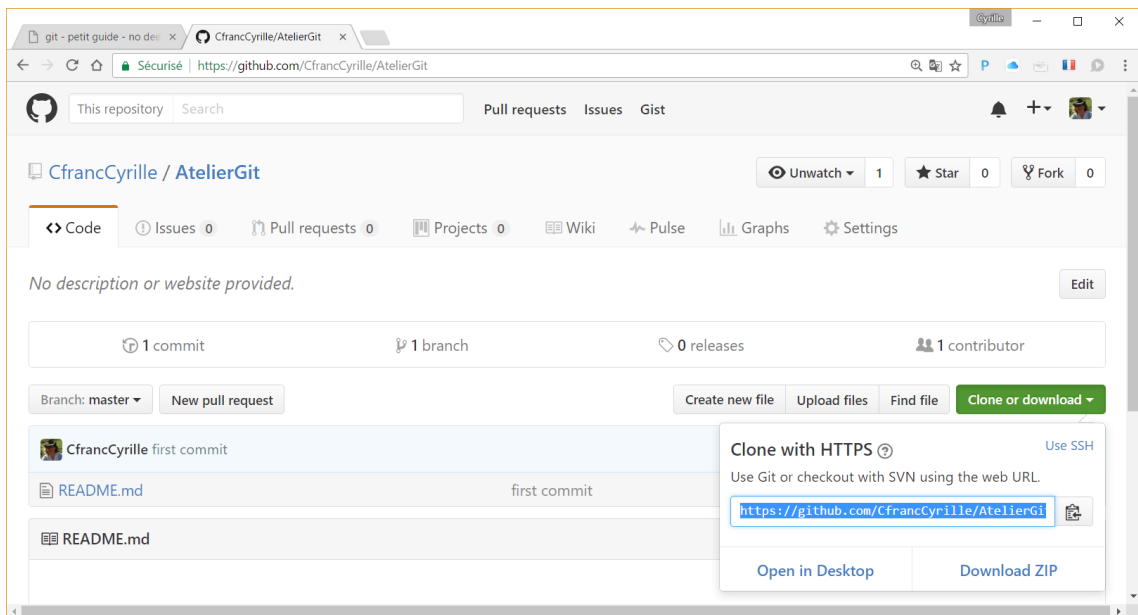
Suivre les différents exercices d'utilisation de Git : Repository du workshop

Indice :

Cloner un projet de github : <https://github.com/collab-uniba/socialcde4eclipse/wiki/How-to-import-a-GitHub-project-into-Eclipse>⁴

On importe le projet si c'est un projet Eclipse : Import all existing Eclipse project after clone finishes

4 - <https://github.com/collab-uniba/socialcde4eclipse/wiki/How-to-import-a-GitHub-project-into-Eclipse>



Trouver et cloner un projet de github

Cloner localement le projet avec git bash :

`git clone <git@ADRESSE_DU_GIT_DISTANT>`

```

MINGW64:/c/Users/Cyrille/git

Cyrille@DESKTOP-DSUSFF2 MINGW64 ~/git (master)
$ cd ..

Cyrille@DESKTOP-DSUSFF2 MINGW64 ~
$ cd git

Cyrille@DESKTOP-DSUSFF2 MINGW64 ~/git (master)
$ git clone https://github.com/CfrancCyrille/AtelierGit.git
Cloning into 'AtelierGit'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

Cyrille@DESKTOP-DSUSFF2 MINGW64 ~/git (master)
$ |
  
```

git clone

Question 3

Créer sa page de profil dans une branche avant de *merger* avec la branche *DEV* :

1. Récupérer le projet : **git pull**
2. Créer une branche pour vos modifications : **git branch <login>** ... (voir indice)
3. Ajouter votre fichier de profil : *profils/<login>.html*
4. *commiter* dans votre repository : *git commit and push*
5. Une fois fini pousser sur la branche **DEV** ... (voir indice)
6. Résoudre les conflits et commiter : *git commit and push*

Attention, il vous faut être ajouté au projet pour pouvoir *push* une modification !

Indices :

Une fois votre branche locale créée on se place dessus et on peut modifier sereinement :

- Se placer dans sa branche : **git checkout <login>**
- Visuellement vous devez être averti de la branche courante !
- Sinon faire un : *git status*

Une fois fini pousser sur la branche DEV :

Attention d'être certain que la branche de DEV est à jours : **git pull**

Se placer sur la branche de DEV : **git checkout DEV**

Merger votre branche **<login>** dans la branche DEV : **git merge <login>**

Question 4

Intégrer le projet dans la branche INT :

1. Se placer sur la branche d'intégration : **git checkout INT**
2. Récupérer le projet : **git pull**
3. Merger la branche DEV dans la branche INT : **git merge <DEV>**
4. Résoudre les conflits et commiter : *git commit and push*
5. Tester l'application...
6. Si il y a des dysfonctionnements... créer des fiches de bug et revenir dans la branche de développement !

Question 5

Pour aller plus loin et s'exercer ;

- Très visuel : <http://learngitbranching.js.org/>⁵
- Un bon approfondissement sans installation nécessaire : <https://try.github.io/levels/1/challenges/1>⁶

Indice :

Revenir à une précédente version : **git reset --hard**

5 - <http://learngitbranching.js.org/>

6 - <https://try.github.io/levels/1/challenges/1>