

GIT BEGINS

JOB 1

Git a bien été installé sur ma vm et je me suis créé un compte github.

JOB 2

Après avoir créé un repository dans github, j'ajoute un dépôt distant avec mon terminal grâce à git clone.

Puis j'initialise mon dépôt vide avec git init qui me crée un sous répertoire .git.

```
sophie@debian:~$ git clone https://github.com/sophie-graff-paille/shell.git
Clonage dans 'shell'...
warning: Vous semblez avoir cloné un dépôt vide.
sophie@debian:~$ git init
astuce: Utilisation de 'master' comme nom de la branche initiale. Le nom de la b
ranche
astuce: par défaut peut changer. Pour configurer le nom de la branche initiale
astuce: pour tous les nouveaux dépôts, et supprimer cet avertissement, lancez :
astuce:
    git config --global init.defaultBranch <nom>
astuce:
astuce: Les noms les plus utilisés à la place de 'master' sont 'main', 'trunk' e
t
astuce: 'development'. La branche nouvellement créée peut être renommée avec :
astuce:
    git branch -m <nom>
astuce:
Dépôt Git vide initialisé dans /home/sophie/.git/
```

Dans l'arborescence je vais dans mon dossier shell. Avec git add j'ajoute mon travail en PDF que j'ai téléchargé avant et mis dans mon dossier shell sur ma vm.

Et je le commit. Mais avec le -m j'aurais dû mettre le message "shell project report"...

```
sophie@debian: ~/shell
sophie@debian:~$ cd shell
sophie@debian:~/shell$ git add SHELL.pdf
sophie@debian:~/shell$ git commit -m SHELL.pdf
[master (commit racine) a4f4f78] SHELL.pdf
Committer: sophie <sophie@debian>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instructions dans votre éditeur pour éditer votre fichier de configuration :

    git config --global --edit

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec
:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 SHELL.pdf
```

Je transfère mon commit de mon dépôt local vers mon dépôt distant avec git push.

```
sophie@debian: ~/shell
sophie@debian:~/shell$ git status
Sur la branche master
Votre branche est basée sur 'origin/master', mais la branche amont a disparu.
(utilisez "git branch --unset-upstream" pour corriger)

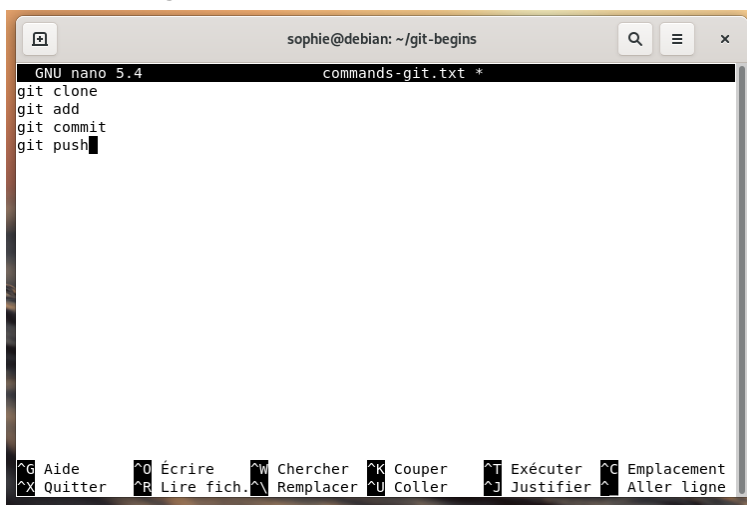
sophie@debian:~/shell$ git push https://github.com/sophie-graff-paille/shell.git
Username for 'https://github.com': sophie-graff-paille
Password for 'https://sophie-graff-paille@github.com':
Énumération des objets: 3, fait.
Décompte des objets: 100% (3/3), fait.
Compression par delta en utilisant jusqu'à 2 fils d'exécution
Compression des objets: 100% (2/2), fait.
Écriture des objets: 100% (3/3), 1.46 Mio | 1.98 Mio/s, fait.
Total 3 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
To https://github.com/sophie-graff-paille/shell.git
 * [new branch] master -> master
sophie@debian:~/shell$
```

JOB 3

Sur github je crée un nouveau repository "git-begins", que je clone et que j'initialise sur ma vm. Je crée également un fichier "commands-git.txt" dans mon dossier "git-begins".

```
sophie@debian:~/shell$ cd
sophie@debian:~$ git clone https://github.com/sophie-graff-paille/git-begins.git
Clonage dans 'git-begins'...
warning: Vous semblez avoir cloné un dépôt vide.
sophie@debian:~$ git init
Dépôt Git existant réinitialisé dans /home/sophie/.git/
sophie@debian:~$ sudo touch commands-git.txt > git-begins
bash: git-begins: est un dossier
sophie@debian:~$ cd git-begins/
sophie@debian:~/git-begins$ sudo touch commands-git.txt
[sudo] Mot de passe de sophie :
sophie@debian:~/git-begins$ sudo nano commands-git.txt
sophie@debian:~/git-begins$
```

Avec la commande nano j'écris les noms des commandes demandées, dans le fichier "commands-git.txt".



```
sophie@debian: ~/git-begins
GNU nano 5.4      commands-git.txt *
git clone
git add
git commit
git push
^G Aide  ^O Écrire  ^W Chercher  ^K Couper  ^T Exécuter  ^C Emplacement
^X Quitter  ^R Lire fich.  ^N Remplacer  ^U Coller  ^J Justifier  ^_ Aller ligne
```

J'ajoute le contenu du fichier "commands-git.txt" avec git add.

```
sophie@debian:~/git-begins$ git add commands-git.txt
```

Je recommence les opérations git commit -m (mais sans message again !! My apologies !)

et git push comme dans le job 2.

```
sophie@debian:~/git-begins$ git commit -m commands-git.txt
[master (commit racine) 32b2e6a] commands-git.txt
Committer: sophie <sophie@debian>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instructions dans votre éditeur pour éditer votre fichier de configuration :

    git config --global --edit

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec :

    git commit --amend --reset-author

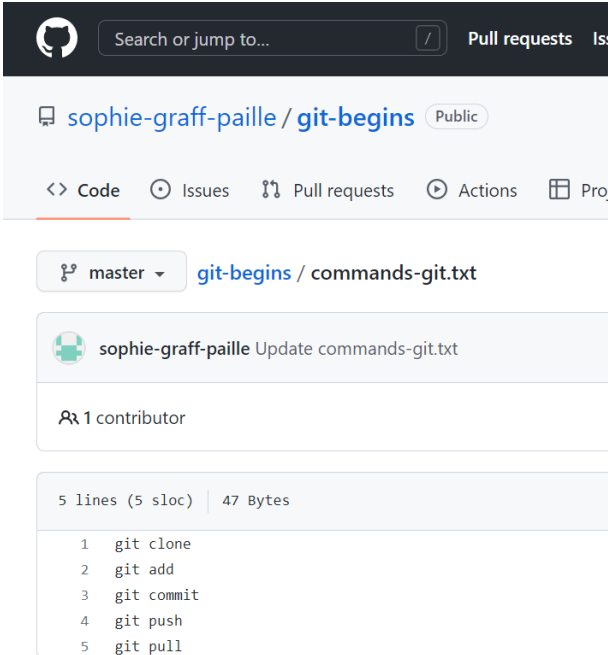
1 file changed, 4 insertions(+)
create mode 100644 commands-git.txt
```

```
sophie@debian:~/git-begins$ git push https://github.com/sophie-graff-paille/git-begins.git
Username for 'https://github.com': sophie-graff-paille
Password for 'https://sophie-graff-paille@github.com':
Énumération des objets: 3, fait.
Décompte des objets: 100% (3/3), fait.
Écriture des objets: 100% (3/3), 244 octets | 244.00 Kio/s, fait.
Total 3 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
To https://github.com/sophie-graff-paille/git-begins.git
 * [new branch]      master -> master
```

Je télécharge et je mets à jour avec git pull.

```
sophie@debian:~/git-begins$ git pull https://github.com/sophie-graff-paille/git-begins.git
astuce: Tirer sans spécifier comment réconcilier les branches divergentes
astuce: est découragé. Vous pouvez éliminer ce message en lançant une des
astuce: commandes suivantes avant votre prochain tirage :
astuce:
astuce:   git config pull.rebase false  # fusion (stratégie par défaut)
astuce:   git config pull.rebase true   # rebase
astuce:   git config pull.ff only        # avance rapide seulement
astuce:
astuce: Vous pouvez remplacer "git config" par "git config --global" pour que
astuce: ce soit l'option par défaut pour tous les dépôts. Vous pouvez aussi
astuce: passer --rebase, --no-rebase ou --ff-only sur la ligne de commande pour
astuce: remplacer à l'invocation la valeur par défaut configurée.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Dépaquetage des objets: 100% (3/3), 675 octets | 225.00 Kio/s, fait.
Depuis https://github.com/sophie-graff-paille/git-begins
 * branch      HEAD      -> FETCH_HEAD
Mise à jour 32b2e6a..41b52a0
Fast-forward
 commands-git.txt | 1 +
 1 file changed, 1 insertion(+)
```

Je vois sur github dans ma branche Master, le contenu de "commands-git.txt".



The screenshot shows a GitHub interface for a pull request. At the top, there's a search bar and navigation links like 'Pull requests' and 'Issues'. Below that, the repository 'sophie-graff-paille / git-begins' is identified as 'Public'. The pull request is for the file 'commands-git.txt' on the 'master' branch. The commit message is 'sophie-graff-paille Update commands-git.txt'. It shows '1 contributor'. The file details are '5 lines (5 sloc)' and '47 Bytes'. The content of the file is displayed as a list of git commands: 1 git clone, 2 git add, 3 git commit, 4 git push, 5 git pull.

Je vérifie sur ma vm

avec nano.

```
sophie@debian:~/git-begins$ nano commands-git.txt
```



```
sophie@debian: ~/git-begins
GNU nano 5.4 commands-git.txt
git clone
git add
git commit
git push
git pull
```

POUR ALLER PLUS LOIN

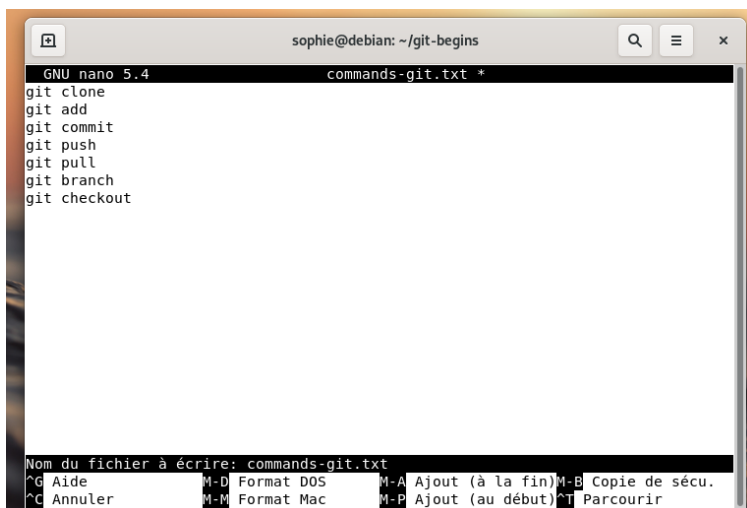
Avec la git branch, sur ma vm, j'ai créé une nouvelle branche "add-create-branch-command" dans mon dossier "git-begins". A nouveau je tape git branch et je vois mes deux branches apparaître.

```
sophie@debian:~/git-begins$ git branch add-create-branch-command
sophie@debian:~/git-begins$ git branch
add-create-branch-command
* master
sophie@debian:~/git-begins$
```

La commande git checkout permet de switcher de branche.

```
sophie@debian:~/git-begins$ git checkout add-create-branch-command
Basculement sur la branche 'add-create-branch-command'
sophie@debian:~/git-begins$
```

Avec nano, je rajoute les deux clés demandées sur mon fichier "commands-git.txt".



```
sophie@debian: ~/git-begins
GNU nano 5.4 commands-git.txt *
git clone
git add
git commit
git push
git pull
git branch
git checkout
```

Comme précédemment, git add, git commit -m (oui je sais toujours sans message!!!) et git push.

```
sophie@debian:~/git-begins$ git add commands-git.txt
sophie@debian:~/git-begins$ git commit -m commands-git.txt
[add-create-branch-command 374be30] commands-git.txt
Committer: sophie <sophie@debian>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instructions dans votre éditeur pour éditer votre fichier de configuration :

    git config --global --edit

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec
:

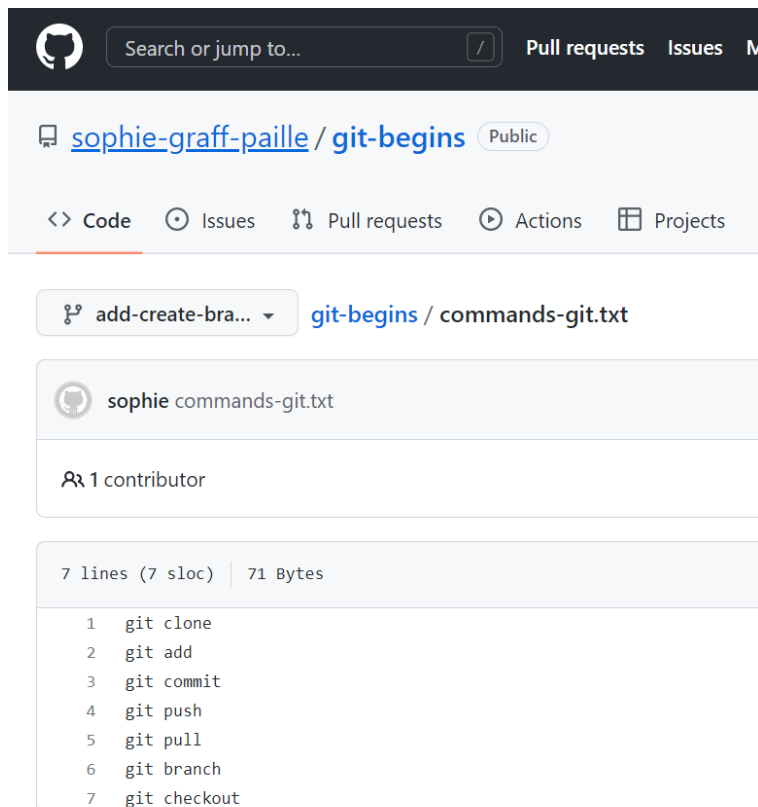
    git commit --amend --reset-author

1 file changed, 2 insertions(+)
```

```
sophie@debian:~/git-begins$ git push https://github.com/sophie-graff-paille/git-
begins.git
Username for 'https://github.com': sophie-graff-paille
Password for 'https://sophie-graff-paille@github.com':
Énumération des objets: 5, fait.
Décompte des objets: 100% (5/5), fait.
Compression par delta en utilisant jusqu'à 2 fils d'exécution
Compression des objets: 100% (2/2), fait.
Écriture des objets: 100% (3/3), 292 octets | 292.00 Kio/s, fait.
Total 3 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
remote:
remote: Create a pull request for 'add-create-branch-command' on GitHub by visit
ing:
remote:      https://github.com/sophie-graff-paille/git-begins/pull/new/add-crea
te-branch-command
remote:
To https://github.com/sophie-graff-paille/git-begins.git
 * [new branch]      add-create-branch-command -> add-create-branch-command
```

Je mets à jour avec git pull.

```
sophie@debian:~/git-begins$ git pull https://github.com/sophie-graff-paille/git-
begins.git
astuce: Tirer sans spécifier comment réconcilier les branches divergentes
astuce: est découragé. Vous pouvez éliminer ce message en lançant une des
astuce: commandes suivantes avant votre prochain tirage :
astuce:
astuce:   git config pull.rebase false  # fusion (stratégie par défaut)
astuce:   git config pull.rebase true   # rebasage
astuce:   git config pull.ff only       # avance rapide seulement
astuce:
astuce: Vous pouvez remplacer "git config" par "git config --global" pour que
astuce: ce soit l'option par défaut pour tous les dépôts. Vous pouvez aussi
astuce: passer --rebase, --no-rebase ou --ff-only sur la ligne de commande pour
astuce: remplacer à l'invocation la valeur par défaut configurée.
Depuis https://github.com/sophie-graff-paille/git-begins
 * branch            HEAD      -> FETCH_HEAD
Déjà à jour.
```



Mes ajouts de commandes n'apparaissent que sur la branche "add-create-branch-command".

Donc j'ai switché sur la branche Master et fait un git merge pour fusionner mes ajouts.

```

sophie@debian:~/git-begins$ git checkout maste
error: le spécificateur de chemin 'maste' ne correspond à aucun fichier connu de
git
sophie@debian:~/git-begins$ git checkout master
Basculement sur la branche 'master'
Votre branche est basée sur 'origin/master', mais la branche amont a disparu.
(utilisez "git branch --unset-upstream" pour corriger)
sophie@debian:~/git-begins$ git merge add-create-branch-command
Mise à jour 41b52a0..374be30
Fast-forward
 commands-git.txt | 2 ++
 1 file changed, 2 insertions(+)
sophie@debian:~/git-begins$ git status
Sur la branche master
Votre branche est basée sur 'origin/master', mais la branche amont a disparu.
(utilisez "git branch --unset-upstream" pour corriger)

rien à valider, la copie de travail est propre
sophie@debian:~/git-begins$ git add commands-git.txt

```

J'ai ensuite recommencé avec git add, git commit -m (...) et git push.

```


sophie@debian:~/git-begins$ git commit -m commands-git.txt
Sur la branche master
Votre branche est basée sur 'origin/master', mais la branche amont a disparu.
(utilisez "git branch --unset-upstream" pour corriger)

rien à valider, la copie de travail est propre
sophie@debian:~/git-begins$ git push https://github.com/sophie-graff-paille/git-
begins
Username for 'https://github.com': sophie-graff-paille
Password for 'https://sophie-graff-paille@github.com':
Total 0 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
To https://github.com/sophie-graff-paille/git-begins
 41b52a0..374be30 master -> master
sophie@debian:~/git-begins$

```

Après la fusion, mes ajouts apparaissent bien sur ma branche Master.

master ▾ [git-begins](#) / commands-git.txt

 **sophie** commands-git.txt

 1 contributor

7 lines (7 sloc) | 71 Bytes

```
1  git clone
2  git add
3  git commit
4  git push
5  git pull
6  git branch
7  git checkout
```

Et pour réparer mon erreur, je vais télécharger ce document en pdf et vous l'envoyer par ma vm dans github comme pour mon dossier shell, en me servant correctement de la commande `git commit -m !!!`

Sophie Graff-Paillé