

# GIT Begins

*Push it to the limit*



## Contexte

---

Vous avez réalisé cette semaine le projet SHELL, il est maintenant temps de recevoir vos premiers points de compétences en **système**. yeahhh 🕶 !

Afin de vous évaluer, l'équipe pédagogique a besoin d'analyser votre travail et donc vous devez le partager. Pour cela vous allez découvrir un outil très répandue dans le monde du développement informatique : **GIT**

## Job 01

---

Avant tout, il faut **installer GIT** sur votre machine. Ouvrir un terminal et écrire :

- **sudo apt install git-all**

Une fois l'outil installé, créez un compte sur [github](#) avec votre **adresse la plateforme** ! il vous faut choisir obligatoirement un username au format **prenom-nom** (votre prénom, un tiret et votre nom, le tout en minuscule !). Respectez le format est **indispensable** pour que l'équipe pédagogique puisse récupérer votre travail.

Avec votre adresse mail la plateforme vous disposez aussi du **compte premium github education**. Une fois connecté, demandez à avoir une upgrade vers [github pour education](#).

La prochaine étape est de mettre sur votre profil Intranet votre username Github, afin que l'équipe pédagogique puisse y avoir accès. Vous devez aussi le **communiquer** via le chat à **Deepthought**, envoyez lui un lien vers votre compte github. Laissez lui quelques instants, il va vous envoyer, au plus vite, une demande par mail vous permettant de rejoindre l'organisation GitHub de La Plateforme (<https://github.laplateforme.io>). En attendant d'être ajouté à l'organisation, vous pouvez continuer votre travail.

## Job 02

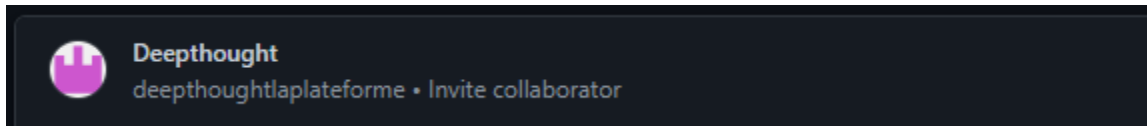
---

**Avant d'avancer, faites une sauvegarde de tout votre travail.**

Dans ce job vous allez rendre votre premier travail. Pour cela :

Sur github :

- créez un repository nommé **shell**
- donnez les droits à **deephoughtlaplateforme** (Attention de donner les droits au bon)



Sur votre terminal en utilisant les lignes de commandes:

- **Clonez** le repository shell
- **Ajoutez** le travail du projet réaliser à l'intérieur du repository shell
- **Pushez** le nouveau contenu de votre repository. *(On prend les bonnes habitudes dès le début, il faut mettre un message de commit explicite ex: "Shell project report")*

Félicitations ! Vous venez de rendre votre premier projet avec GIT.

## Job 03

---

De retour sur github créer un nouveau repo nommé **git-begins** de la même manière **cloner** ce dernier via le **terminal**.

- Dans le terminal et dans le repo cloner, **créer** un fichier nommé **"commands-git"**, dans ce fichier écrivez **les commandes git** qui permettent de :

- Cloner un repo
- Les 3 commandes de push (dans l'ordre)

Réalisez un **push**.

Sur github, modifier le contenu du fichier **"commands-git"** en y ajoutant la commande qui permet de **pull**.

De retour sur votre terminal, réalisez la commande de pull sur votre repository git-begins. Vous devriez y voir dans le fichier *"commands-git"* les modifications effectuées sur github.

Vous avez réussi ? Parfait !

A présent, créez sur Github un repo nommé 'Starter-2022'. Mettez y tout votre dossier prenom.nom présent dans votre drive, et pushez le tout. Vérifiez que votre repos se nomme bien comme ci-dessus (majuscules comprises), et que vous l'avez bien partagé avec deepthought. Et voilà, tout votre travail depuis le début de l'année est en ligne, versionné, et disponible sur github. Bien joué !

## Pour aller plus loin...

---

Dans le repo **git-begins**, sur le **terminal** et toujours en **ligne de commande** créer une branche **"add-create-branch-command"**. **Basculer** sur cette branche.

Ajouter dans le fichier **"commands-git"** la commande permettant de créer une branche ainsi que celle qui permet de switcher de branche. Réalisez un push.

Retournez sur votre branche principale (main), et réalisez un merge de la branche créée précédemment, afin d'avoir le fichier commands-git. Si vous l'avez, vous pouvez retourner sur la branche secondaire. Ouvrez le fichier 'commands-git', et effacez le contenu, afin d'y écrire autre chose. Faites un commit, puis un push. Retournez sur votre branche principale, et faites un git merge de votre branche secondaire.

Bravo, vous avez votre premier conflit de merge. A vous de le résoudre.

Ensuite suivez les étapes suivantes :

- Faites une sauvegarde de votre travail en local
- vérifiez que vous soyez bien sur votre branche principale

- Supprimez votre travail (sauvegardé en amont, dans un autre dossier en local)
- Faites un commit, puis un push
- Une fois cela fait, allez à la page suivante

- Bravo, vous venez de perdre tout votre travail. Pas de panique, vous avez fait une sauvegarde en local (heureusement)
- Faites en sorte de le récupérer votre travail via des commandes git, puis de le repush sur la branche principale

## Rendu

---

Le projet est à rendre sur <https://github.com/prenom-nom/git-begins> et <https://github.com/prenom-nom/Starter-2022>  
Pensez à donner les droits sur le répertoire à **deephoughtlaplateforme** !

## Compétences visées

---

- Git

## Base de connaissances

---

- Github - Site officiel
- Qu'est-ce que Git ?
- GIT - Le site