

# git : créer un premier projet (dépôt)

## Lancer la console Git

- Lance le Shell de Git (Git bash) :
- tu obtiens la fenêtre suivante :

## Configuration de base de Git

La configuration de base consiste à préciser :

- ton **nom d'utilisateur**,
- ton **adresse de courriel** qui doit être la même que celle qui te permet de d'identifier sur le site Github,
- l'**activation des couleurs** lors de l'affichage des informations du dépôt.

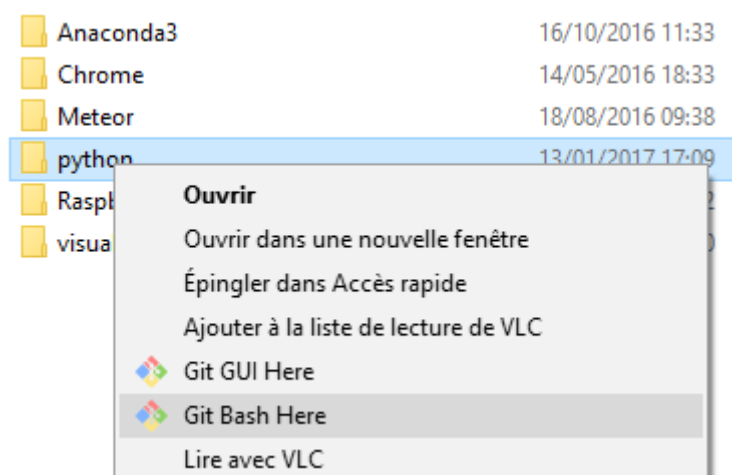
Les configurations de git sont soit **globales**, soit **propres à chaque dépôt**. En pratique, si une valeur de paramètre n'est pas trouvée au sein du dépôt, git va regarder la configuration globale.

Voici les paramètres minimum à préciser :

```
git config --global user.name "Votre nom"
git config --global user.email "adresse@email.com"
```

## Créer un premier dépôt local vide

- positionne-toi dans le dossier où tu enregistres tes programmes python :
- autre solution : depuis l'explorateur, clique-droit sur le dossier où tu souhaites créer ton dépôt et choisis le shell bash (Git Bash):



- lance la commande de **création d'un dépôt local** :

```
$ git init projet
```

Ton dépôt local est créé. En effet le dossier **projet** a été créé mais également un **dossier caché .git** qu'il ne faut pas supprimer ni modifier car il permet à Git de gérer ton projet.

## Première gestion du dépôt

- entre dans le dossier de ton dépôt local

```
$ cd projet
```

- lance **Spyder** et crée ton programme Python en l'enregistrant dans ce **dossier projet**. Comme exemple, c'est le fichier `lancerdes.py` qui a été créé.
- reviens sur la console de Git. La commande **ls** permet de visualiser le nouveau fichier :

Le fichier qui vient d'être créé dans ton dossier **Projet** n'est pas encore pris en compte par Git comme faisant partie du dépôt du projet. Il est pour considéré comme un **fichier de travail** et non comme le fichier définitif à mettre dans le dépôt du projet.

## Vérification de l'état des fichiers

- lance la commande **git status** :

Le fichier que tu viens d'ajouter est **listé en rouge** comme étant **Untracked files** (Fichiers non suivis). Cela signifie que le fichier que tu viens d'ajouter dans le dossier **projet** n'est pas encore en suivi de version par Git.

- pour **activer le suivi de version** et indexer le fichier, tape la commande **git add lancerdes.py** puis la commande **git status** :

Le nom du fichier est maintenant affiché en vert ce qui signifie qu'il est maintenant **suivi et indexé par Git**.

Un fichier indexé apparaît dans la section **Changes to be committed**. Cela signifie qu'il est **prêt à être validé**, c'est à dire prêt à mettre mis dans le dépôt comme version définitive.

- Modifie le contenu du fichier et relance la commande **git status** :

Git a détecté que le fichier a été modifié :

- il apparaît dans la section **Changes not staged for commit**,
- ton fichier est cependant toujours présent dans la section **Changes to be committed**.

Quand un fichier apparaît à la fois dans la section **Changes to be committed** et dans la section **Changes not staged for commit** cela signifie que tu peux valider la version qui est indexée, mais pas la dernière mise jour.

Pour **indexer la dernière mise à jour** du fichier et donc pouvoir valider la toute dernière version de ton fichier, tape la commande **git add lancerdes.py** :

- pour valider les fichiers indexés, tape la commande suivant en mettant un commentaire `*git commit -m 'commentaires'` :

Et il n'y a plus de fichier indexé à valider :

Lien : <http://defeo.lu/in202/>

## Retour à l'activité : utiliser git, logiciel de gestion de versions

- [Cours : utiliser git, logiciel de contrôle de version](#) </WRAP>

From:

/ - **Les cours du BTS SIO**

Permanent link:

[/doku.php/icn/facultatif/a\\_git\\_03](#)

Last update: **2017/01/15 00:03**

