

# ☐ Quelques recettes

OK, maintenant que l'on a installé et configuré Git, on peut maintenant passer à comment l'utiliser. On va commencer par les choses les plus communes dans Git.

## Créer un nouveau projet avec Git

Tout d'abord on va créer un nouveau projet Git en créant un nouveau dossier, et en l'initiant avec Git. (`mkdir` permet de créer un dossier, `cd` permet d'aller dedans et `git init` initialise un nouveau projet)

```
mkdir mon-super-projet
cd mon-super-projet
git init
```

Ensuite on va créer un nouveau fichier dans ce dossier, par exemple "README.md"

```
# La commande suivante écrit "Hello World" dans un nouveau fichier "README.md"
echo "Hello World" > README.md
```

Si on veut voir le status actuel de notre projet on peut utiliser les commandes `diff` et `status`

```
# La commande status liste les fichiers modifiés
git status
# La commande diff, liste les changements parmi les fichiers "suivi" (ajouté précédemment dans l'historique)
git diff
# Cette dernière commande peut aussi servir pour avoir les différences entre plusieurs fichiers ou plusieurs moments dans l'historique
```

Une fois que l'on a fait des changements on peut ajouter ces changements dans l'historique Git

```
# Git add ajouter le fichier dans la liste des changements qui vont être ajouté à l'historique
# On peut remplacer le nom du fichier par * (étoile) ou . (point) pour ajouter tous les fichiers et dossiers
git add README.md
# Git commit l'ajoute à l'historique avec un message de description (obligatoire)
git commit -m "J'ai ajouté un nouveau fichier"
```

```
# On peut par après utiliser -am à la place de -m pour ajouter tous les fichiers qui ont déjà été commit
```

On peut ensuite voir que notre "commit" à été ajouté à l'historique du projet avec `log`

```
# Cette commande liste la liste des commits
```

```
git log
```

# Définir un dossier existant comme projet Git

Tout d'abord on va aller dedans, soit en utilisant `cd` soit en faisant "clic droit" > "ouvrir Git bash ici". Ici je vais utiliser `cd`

```
cd /chemin/de/mon/dossier
```

Ensuite on peut initialiser Git dans le dossier

```
git init
```

Enfin on peut ajouter tout le dossier dans l'historique pour commencer

```
# le "." indique tout le dossier. Mais "*" aurait aussi pu fonctionner pour ajouter tous les fichiers et sous dossiers.
```

```
git add .
```

```
git commit -m "Premier commit"
```

# Utilisation de Git en temps normal

Maintenant imaginons vous faites vos modifications dans vos fichiers, etc. Et vous souhaitez "commit" les changements.

```
# en utilisant le flag -a, on ajoute automatiquement tous les fichiers "suivis"
```

```
# C'est à dire, tous les fichiers qui ont déjà été ajouté dans le passé avec "git add"
```

```
git commit -am "J'ai modifié des trucs"
```

```
# Ou, si on veut tout envoyer d'un coup
```

```
git add .
```

```
git commit -m "J'ai ajouté des trucs"
```

```
# Ou encore si on veut faire fichier par fichier
```

```
git add a.txt b.java
```

```
git commit -m "J'ai ajouté des trucs"
```

```
# Envoyer les changements sur le serveur une fois satisfait
```

```
# on va voir cela plus en détails dans la section suivante
```

```
git push origin master
```

```
# Note: remplacer "master" par la branche concernée
```

---

Revision #1

Created 26 April 2023 19:09:19 by SnowCode

Updated 26 April 2023 19:09:20 by SnowCode