

Git Cheat Sheet

[blabla] : optionnel

<blabla>: obligatoire

Config

Permet de configurer GIT pour dire quel compte l'utilise.

```
git config --global user.name "Your Name"
git config --global user.email "youremail@yourdomain.com"
```

Clone

Un clone permet de récupérer un projet depuis un repo distant. Pour cela il faut prendre le lien HTTPS de celui-ci.

```
git clone [-b branch_name] <repo_link>
```

Il est obligatoire d'ajouter le lien du repo (évidemment). Et on peut préciser la branche à cloner si besoin.

Status

Permet de voir l'état du repo (les pull à faire, les fichiers modifiés non commité, les commits, etc...).

```
git status
```

Diff

Permet de voir les changements effectués en local. Sur tous les fichiers, ou un fichier spécifique, voir un dossier spécifique.

```
git diff [file/directory_name]
```

Add

Permet d'ajouter (ou *stage*) un(des) fichier(s) au prochain commit.

```
git add <file_name>
```

```
git add .
```

En précisant le nom du fichier, on ne rajoute qu'un seul fichier.

En mettant `.` on ajoute automatiquement tout les fichiers modifiés.

Commit

Permet de créer un commit avec les fichiers précédemment *stage*, avec un message de commit.

```
git commit -m <commit_message>
```

Pull

Permet de récupérer les changements, sur la branche actuelle, venant directement du repo distant.

```
git pull
```

Si conflits détectés, cela vous sera signaler, il faudra les régler sur le code, puis reprendre depuis le **add**.

Push

Permet d'envoyer le/les commit(s) au repo distant, directement sur la branche en cours.

```
git push
```

Automatic Merge

Quand un conflit est détecté, mais que celui-ci se résout automatique, un merge sera créer pour vous, et cela demandera de valider ou créer un message.

Pour laisser le message par défaut:

```
*press ESCAPE*  
:wq
```

Pour éditer le message:

```
i  
<message>  
*press ESCAPE*  
:wq
```

Branch

Permet de lister les branches disponibles:

```
git branch
```

Ou de créer une nouvelle branche:

```
git branch <branch_name>
```

Ou supprimer une branche en local:

```
git branch -d <branch_name>
```

Checkout

Permet de changer de branche:

```
git checkout <branch_name>
```

Ou de créer une branche et aller dessus:

```
git checkout -b <branch_name>
```

Merge

Permet de merger une branche sur la branche actuellement. (une sorte de pull d'une branche vers une autre)

```
git merge <branch_name>
```