

GIT

- [Memo Commandes GIT](#)

Memo Commandes GIT

Gestion contenu

- Ajouter un fichier ou dossier modifié

```
# Add a specific file
git add filename
# Add all changed files in current directory
git add .
```

- Pour créer un nouveau commit:

```
# Add commit message directly in commit command
git commit -m "New commit"
# Open editor to write commit message
git commit
```

- Pousser les derniers commit et fichiers vers le serveur

```
push origin branch_name
# Push to "main" branch
push origin main
```

Gestions des branches

- Lister les branches existante du dépôt

```
git branch
```

- Basculer sur une branche existante

```
git checkout <branch_name>
```

- On peut créer une nouvelle branche avec l'option **-b**

```
git checkout -b <new_branch>
```

Submodule

Un submodule est un repo git à l'intérieur d'un autre dépôt. C'est d'une certaine manière un lien vers un autre repo.

Pour ajouter un dépôt "demo" dans "test"

```
git submodule add url_repo_git:user/demo.git
```

Cette commande doit être exécutée dans un dossier dépôt GIT.

Notre "dossier" submodule pointe vers un commit spécifique (le dernier quand on l'a initialiser). Si sur le dépôt il y a eu de nouveaux commit, ils ne seront pas pris en compte tant qu'on a pas update le submodule.

Pour mettre à jours tous nos submodule et merger les fichiers, se placer à la racine de notre dépôt:

```
git submodule update --remote --merge
```

C'est l'équivalent d'un `git pull` dans notre dossier repot du submodule

Clon d'un dépôt contenant des submodule

Quand on clone un dépôt git avec des sousmodules, c'est derniers ne sont pas initialiser. Pour les mettre à jour:

```
git submodule init  
git submodule update
```

Ce qui initialisera et mettra à jour les fichiers dans les dossiers des sousmodules, avec le commit référencé pour ces derniers.

Liens

<https://wiki-tech.io/Cheatsheet/git>