

Git

Documentation générale : <https://gist.github.com/aquelito/8596717>

Petite guide pour bien démarrer : <http://rogerdudler.github.io/git-guide/index.fr.html>

Avec un logiciel : SourceTree

En ligne de commande :

Setup :

```
$ git config --global user.name "Mickael Masset"  
$ git config --global user.email "adresse@email.com"
```

Créer un repository local a partir d'un dépôt distant (par ex: sur gitlab.com) :

```
$ git clone git@gitlab.com:nom-compte/nom-projet.git // Crée le dossier du projet et le clone
```

Créer un nouveau fichier dans mon repository pour faire une mise à jour :

```
$ cd nom-projet/ // Aller dans le projet  
$ touch README.md // Créer un fichier readme (facultatif)  
$ git add README.md // Envoyer un seul fichier  
$ git commit -m "ajout du fichier README" // Ajouter un commentaire a mon commit  
$ git push -u origin master // -u pour configurer la branche
```

Infos d'un projet GIT :

```
$ git status // Status des fichiers
```

Commit :

```
$ git add . // Pour tout mettre à jour (ou chemin d'un seul fichier)  
$ git commit -m "message_du_commit" // Pour commenter mon commit
```

Mettre à jour le dépôt local (pour vérifier les mises à jours avant de commit)

```
$ git pull
```

Envoyer son commit (mise à jour vers le Git)

```
$ git push
```

Créer un fichier **.gitignore** pour ignorer les sauvegardes de certains fichiers

```
# Ignorer tous les fichiers dans le répertoire  
bin bin/  
# Ignorer tous les fichiers commençant par .  
.*
```

Gérer les branches

```
$ git branch // Liste des branches  
$ git branch nom_de_branche // Créer une branche  
$ git checkout nom_de_branche // On se positionne sur la branche pour travailler dessus  
$ git checkout -b nom_de_branche // en une seule commande : on crée et on se positionne sur la branche
```

Fusion de branche *Merge*

Les fusions s'exécutent toujours en local, sur des branches validées.

Pour fusionner 2 branches, on se place sur la branche principale (celle que l'on veut garder)

/!\ Git peut vous demander de régler des conflits entre une même page mise à jour dans 2 branches

Il est préférable de créer des branches spécifiques par type d'intervention (ex : une branche pour le front, une branche pour le back ... pour éviter les conflits)

```
$ git branch // Liste des branches  
$ git branch nom_de_branche // Créer une branche  
$ git checkout nom_de_branche // On se positionne sur la branche pour travailler dessus  
$ git checkout -b nom_de_branche // en une seule commande : on crée et on se positionne sur la branche
```