**Github**

**Configuration**

Identity Name

$ git config --global user.name "aquelito"

Identity Email

$ git config --global user.email "axel@aquelito.fr"

Editor Tool

$ git config --global core.editor subl

Diff Tool

$ git config --global merge.tool filemerge

Liste des globals

$ git config --list

**Principales commandes**

Status des fichiers

$ git status

Lister les branchs

$ git branch

\*sur la branche courante.

Créer une branch

$ git branch nom\_de\_ma\_branch

Changer de branch

$ git checkout nom\_de\_ma\_branch

Premier commit

$ git add .

$ git commit - m "initial commit"

Commit suivant

$ git add chemin\_vers\_mon\_fichier

$ git commit -m "message du commit"

Annuler le dernier commit et modifs

$ git reset --hard md5\_commit

$ git push --force

Mettre à jour le dépôt local

$ git pull

Envoyer ses commits vers le dépôt distant

$ git push

Supprimer un fichier du répertoire de travail et de l'index

$ git rm nom\_du\_fichier

Supprimer un fichier de l'index

$ git rmg --cached nom\_du\_fichier

**Diff**

Affiche la différence entre le contenu du dernier commit et celui du répertoire de travail. Cela correspond à ce qui serait commité par git commit -a.

$ git diff HEAD

Affiche la différence entre le contenu pointé par A et celui pointé par B.

$ git diff A B

Diff entre un dossier présent sur deux branches

$ git diff master..MA\_BRANCH chemin/vers/mon\_dossier

**Log**

$ git log

# Affiche X derniers commits

$ git log -n X

# Affiche un ensemble de commits par date

$ git log --since=date --until=date

Représentation de l’historique à partir de HEAD (commit / branch)

$ git log --oneline --graph --decorate

Représentation de l’historique à partir d'un fichier (commit / branch)

$ git log --oneline --graph --decorate nom\_du\_fichier

**Annuler commits (soft)**

Seul le commit est retiré de Git ; vos fichiers, eux, restent modifiés. Vous pouvez alors à nouveau changer vos fichiers si besoin est et refaire un commit.

Annuler le dernier commit

$ git reset HEAD^

Pour indiquer à quel commit on souhaite revenir, il existe plusieurs notations :

* HEAD : dernier commit ;
* HEAD^ : avant-dernier commit ;
* HEAD^^ : avant-avant-dernier commit ;
* HEAD~2 : avant-avant-dernier commit (notation équivalente) ;
* d6d98923868578a7f38dea79833b56d0326fcba1 : indique un numéro de commit précis ;

**Annuler commits (hard)**

Si vous voulez annuler votre dernier commit et les changements effectués dans les fichiers, il faut faire un reset hard. *Cela annulera sans confirmation tout votre travail !*

Annuler les commits et perdre tous les changements

$ git reset --hard HEAD^

*Annuler les modifications d’un fichier avant un commit*

Si vous avez modifié plusieurs fichiers mais que vous n’avez pas encore envoyé le commit et que vous voulez restaurer un fichier tel qu’il était au dernier commit :

$ git checkout nom\_du\_fichier

*Annuler/Supprimer un fichier avant un commit*

Supposer que vous venez d’ajouter un fichier à Git avec git add et que vous vous apprêtez à le « commiter ». Cependant, vous vous rendez compte que ce fichier est une mauvaise idée et vous voulez annuler votre git add.

Il est possible de retirer un fichier qui avait été ajouté pour être « commité » en procédant comme suit :

$ git reset HEAD -- nom\_du\_fichier\_a\_supprimer

[**Raw**](https://gist.github.com/aquelito/8596717/raw/b3ac8dac7e309f589af093c8620025187bfb7019/installer_un_projet.md)

[**installer\_un\_projet.md**](https://gist.github.com/aquelito/8596717#file-installer_un_projet-md)

**Installation**

**Récuperer le repo sur Github**

$ git clone ....

Mon repo est composé d'au moins deux branch.

develop : dédié au développement et résolution de bug. master : reflète le code en production. Personne ne doit travailler directement sur cette branch.

Pour récupérer votre branch develop

$ git branch -a

$ git checkout origin/develop

$ git checkout -b develop origin/develop

$ git branch

**Developpement : Branch develop**

**Prod : Branch Master**

# On se met sur la branche master

$ git checkout master

# On merge la branche develop

$ git merge develop

# Pour pousser les changements

$ git push origin master

# Penser à revenir sur develop

$ git checkout develop