Versionning Git Github

Les outils de versioning

Qu’est ce que c’est qu’un système de contrôle de version :
Permet de faire vivre le code base, le partagé et pouvoir travaillé ensemble. Peut importe le langage c’est compatible. SCM source code management comme Git

Permet de stocker le code et toutes ces versions. On stocke un instantanée (une photo) du projet à l’instant T et pouvoir le mettre dans une arborescence pour pouvoir reprendre à n’importe quelle version du code.

- Traçabilité
- Collaboration
- Peu de conflit et facilité de gestion
- Code fonctionnel et développement simultané
- Création de workflow

Les principaux SCM

Git, AWS Code Commit, Azure DevOps Server, Subversion / SVM (Apache), Mercurial, CVS, ….

> ➡️ Peer to Peer : Torrent → partage/duplique de fichier entre plusieurs pc → Système décentralisé / distribuer → Permet de récupérer des fichiers même si son pc brûle

Repository alias repo : espace disque dans lequel sera stoker et sauvegarder le projet/note

git logiciel de versionning → github app web qui donne une interface graphique et du stockage en cloud

Les mauvaises pratiques

- Projet Web sans SCM
- Utilisation de méthodes “dépassées” ou risquées : OneDrive, Réseau Sociaux, courrier électronique, disque dut
- Très difficile de travailler à plusieurs
- Mise en commun : sources d’erreur
- Perte de temps considérable

Les différentes commandes

`git init` : initialise le dossier git localement (dans un dossier caché git : tout changement est tracé, enregistre les modifications) dossier courant

Un dépôt local est un dossier caché git

`git status`: indique le statut du dépôt

git a chaque version sauvegarde l’intégralité des fichiers du projet. Il est très dure de remonter dans un projet si on enregistre que les changements.

git fonctionne avec des snapshots.

Quand on travaille sur git il créait de base une branche. Les commit / sauvegarde permettent de sauvegarder les fichiers sur le repo.

`git diff` : montre les fichiers modifier

`git log` log command git

`git add fichier` permet de suivre un fichier ou `git add *` ajoute tous les fichiers non suivi

`warning LF will  be replaced by CRLF` : format d’écriture du Reedme

`git commit -m "Nom du commit"` permet de sauvegarder pour git localement pour l’instant

➡️ Opérations locales pour la plupart - Historique entièrement consultable hors connexion

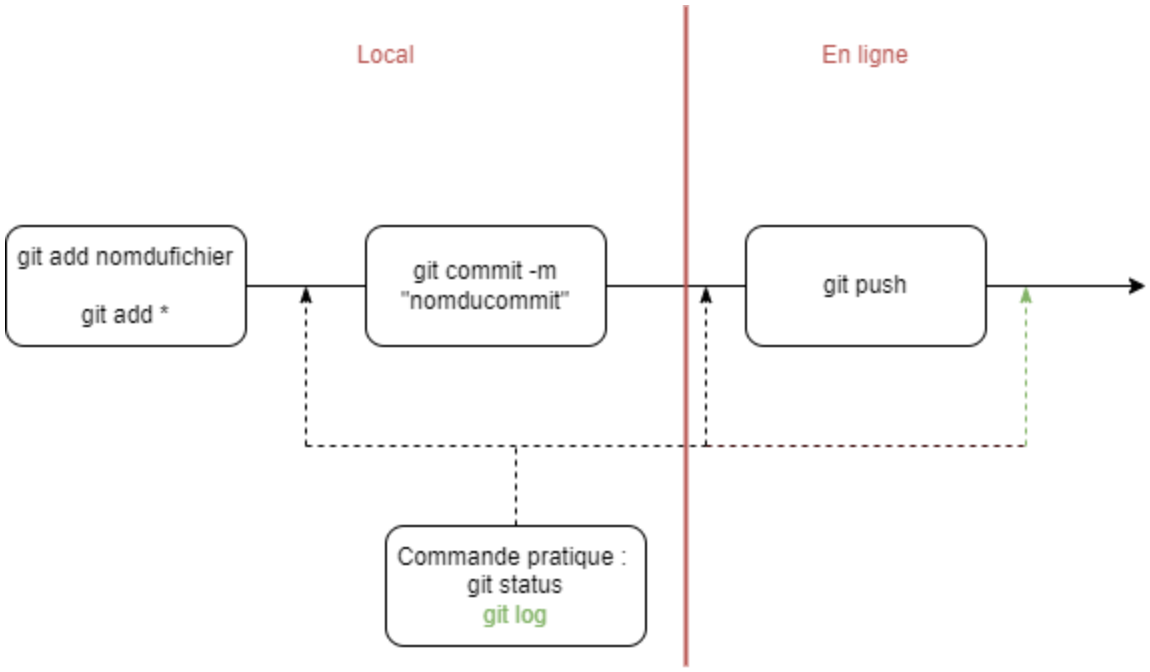
Checkout : récupérer le projet sur github pour le mettre en local

`git branch -M main` change le nom de la branche principale

`git remote add origin https://github.com/DRmatteo74/GithubVersionning.git` : permet à git d'envoyer l'alias origin pour le chemin du projet github. Permet de faire un raccourci vers l'URL du repository distant. Remote : élément du type distant

`git push -u origin main` : permet d'envoyer les commit sur la branche main sur le repo distant. S'assurer que la branche main sur le local doit etre synchroniser avec la même branche sur github

`git push` : permet d'envoyer les fichiers modifier localement en lig



▼ Clé ssh

clé public, clé privé en local pour chiffrer et communiquer. La clé public doit être fournit à github.

`ssh-keygen -t ed25519 -C "mdirienzo74@gmail.com”` créer un clé ssh (public et privé) à mettre dans github.

`eval "$(ssh-agent -s)”` : ajoute la clé (pas obligatoire)

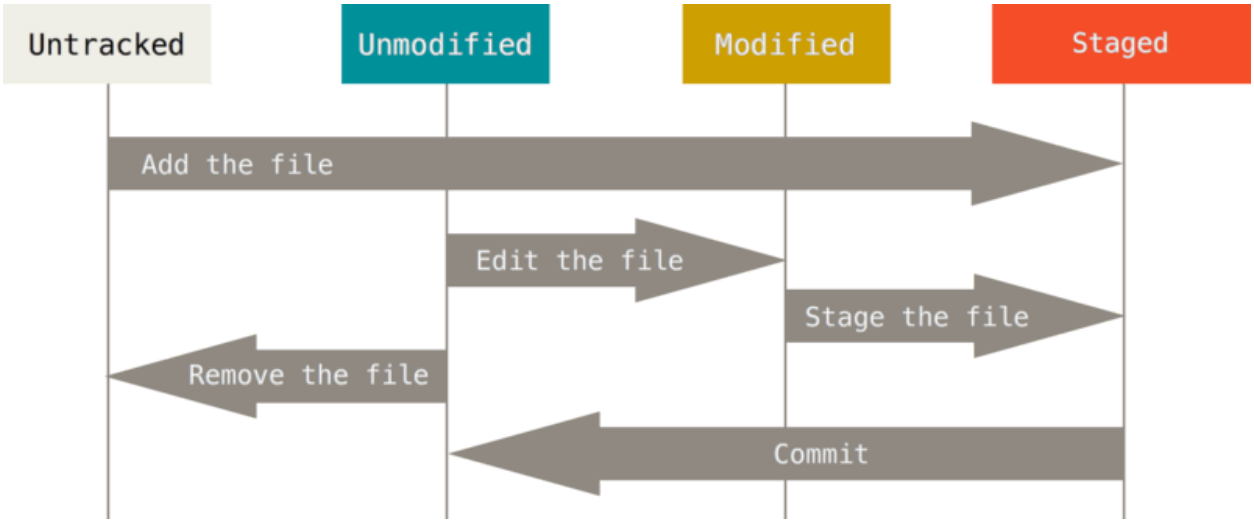
▼ Etats pour les fichiers :

- Modifié (Modify) → Si un fichier est modifier localement
- Indexé (Staged) → Quand les fichiers sont suivi (git add …)
- Validé (Comited) → Quand les fichiers sont sauvegardé localement dans git (git commit …)

`git config -l` : affiche la config du repo git local

`git commit -a`: permet de add et commit tous les fichiers d’un coup

`git commit / git pull / git push`



Décrit les différents états d’un fichier

`rm -rf .git/*` : supprime le repo local

`git clone lienssh` : clone les fichiers en local

`git rm --cached test.txt` : retire le fichier du suivi git. Les modifications ne seront pas envoyé.

Enlever les dépendances et les compilations des programmes non essentiel

Fichier `.gitignore` enlève les fichiers que l'on ne souhaite pas envoyé

`git reset HEAD test.txt` : supprime le fichier du add. HEAD : pointe sur le commit le plus récent

`git mv restore.txt renamed.txt`: renomme le fichier

`git log --pretty=online` : affiche tous les commits pas encore push

Les branches

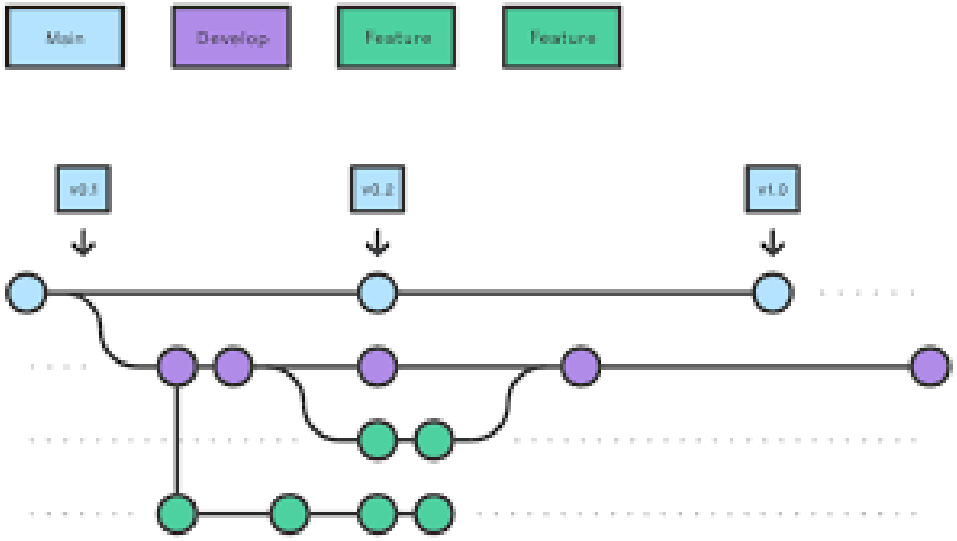
Quand on crée une branche on crée une copie du projet à partir d'un commit donné sans impacter le projet de base.

`git branch nom` : créer une branch local

`git branch` : affiche les branches

`git checkout nom` : change la branche active "-v" permet de créer la branche et changer dans la volée

`git merge` : assemble les branches / fusionne


Main
Develop
Feature
Feature
v0.1
v0.2
v1.0