

Git est un logiciel de gestion de version.

GitHub est un service en ligne d'hébergement de dépôts Git qui fait office de serveur central pour ces dépôts.

Les deux modèles des logiciels de gestion de version : **modèle centralisé** vs **modèle décentralisé**

Modèle centralisé: la source du code du projet est hébergée sur un serveur distant central et les différents utilisateurs doivent se connecter à ce serveur pour travailler sur ce code.

Modèle distribué: le code source du projet est toujours hébergé sur un serveur distant mais chaque utilisateur est invité à télécharger et à héberger l'intégralité du code source du projet sur sa propre machine.

Paramétrage de Git

`git config --global user.name "Pierre Giraud"` : renseigner un nom

`git config --global user.email pierre.giraud@edhec.com` : renseigner une adresse email.

`git config --global core.editor nom_de_votre_editeur`

Démarrer un dépôt Git

- On peut importer un répertoire déjà existant dans Git
- On peut cloner un dépôt Git déjà existant.

Les états des fichiers

Ensuite, chaque fichier suivi peut avoir l'un de ces trois états :

- **Untracked**
- **Unmodified**
- **Modifié ("modified")**
- **Indexé ("staged")**
- **Validé ("committed")**

`git init` : Pour initialiser un dépôt Git

`git status` : pour déterminer l'état des fichiers de notre répertoire

`git add` : Pour indexer des fichiers

`git add *.txt` : index tous les fichiers du projet qui possèdent une extension .txt

`git commit` : la validation / l'enregistrement en base de données.

git add + git commit => git commit -a

git rm : Pour supprimer un fichier et l'exclure du suivi de version

git rm --cached : exclure un fichier du suivi Git mais le conserver dans le projet

git reset HEAD nom-du-fichier : Retirer un fichier de la zone d'index de Git

continuera d'être suivi par Git

ne fera donc pas partie du prochain commit

.gitignore : un fichier qui contient les différents fichiers qu'on souhaite ignorer.

git mv ancien-nom-fichier nouveau-nom-fichier : Renommer un fichier dans Git

git log : Consulter l'historique des modifications Gi

Cette commande affiche la liste des commits réalisés du plus récent au plus ancien. Par défaut, chaque commit est affiché avec sa somme de contrôle SHA-1, le nom et l'e-mail de l'auteur, la date et le message du commit.

git log -p : Permet d'afficher explicitement les différences introduites entre chaque validation.

git log --pretty=oneline :Afficher chaque commit sur une seule ligne

git log --since : Afficher que les modifications depuis une certaine date

git log --author : Afficher que les commits d'un auteur en particulier

git commit --amend :Écraser et remplacer un commit

git checkout -- nom-du-fichier : Annuler des modifications apportées à un fichier

git restore :Annuler des modifications apportées à un fichier.

Une branche :une copie physique de la totalité du répertoire de travail est effectuée

Une branche, dans Git, est simplement un pointeur vers un commit .

La branche par défaut dans Git s'appelle master.

git branch nom-de-la-branche : Créer une nouvelle branche

HEAD : un autre pointeur spécial qui pointe sur la branche master par défaut.

git checkout : basculer à une autre branche.

git branch

git checkout => git checkout -b : pour créer une branche et basculer immédiatement dessus.

Pour fusionner nos deux branches:

git checkout master :on va se placer sur master

git merge test : test c'est le nom de la branche qu'on souhaite fusionner avec master.

git branch -d : effacer une branche.

git rebase :