

Tuto utilisation Git

Qu'est ce que Git ?

Git est un logiciel de gestion de versions décentralisé. Cela est utilisé dans le cadre d'un projet en groupe pour mieux gérer les conflits de version et d'effectuer un travail collaboratif de bonne qualité.

mon gars sur:

https://www.youtube.com/watch?v=zNxGgl6O5NE&list=PLlxQJeQRaKDRBd_Fazel7gLq5wyr7f7J

Installation

- linux
sudo apt-get install git
- windows
site git: <https://git-scm.com/>

Il est également nécessaire de créer un compte sur gitlab ou github et d'ajouter la clé rsa publique du pc.

Configuration

Après avoir installé git, il est nécessaire de le configurer afin de pouvoir s'identifier lorsque nous voulons poster ou corriger un travail sur les plateformes type github ou gitlab.

git config

- obtenir une liste des paramètres de configuration

git config --global user.name "<Prenom+Nom>"

- Identifier qui a fait le commit.

git config --global user.email "<votre email>"

I. Quand on crée un nouveau dépôt de travail

On va sur la plateforme (github/gitlab), on suit les instructions.

II. Clonage

Quand le dépôt de travail existe déjà et qu'on veut ajouter notre contribution, il suffit de cloner le dépôt.

git clone <lien>

III. Commande important

git init : - initialise le projet avec un dépôt git.

git status : - connaître le statut actuel du dépôt

git add <file> ou git add <file>

- commencer a traquer un fichier
- -A pour tous traquer dans le répertoire

(S'il y a des fichiers qu'on ne souhaite pas traquer on les ajoute dans le fichier ".gitignore".)

git commit -m "nom du commit"

- Quand on sent que le travail est bon, on fe un backup, -m pour le message qui decrit le commit

git log

- voir tous les commits qu'on a fait

git diff <fichier>

- connaître les éléments ayant été modifié dans le fichier.

git checkout -f

- si on a supprimer un fichier par inadvertance ou autres, on peut les récupérés (sorte de ctrl -d).

Pull

git pull : - Permet de mettre a jour un depot en comptabilisant les modification effectuées par les autres personnes.

Push

git push : - Envoyer sur le platform notre travail.

Système de branchement

Lorsqu'on travail sur une nouvele fonctionnalite il est conseiller de cree une branche, et lorsqu'on pense que le travail est satisfaisant on peut revenir sur la brache principal ui est la branche "master".

git branch

- avoir une vision de l'ensemble de nos branche. L'etoire "*" désigne la branche ou l'on est.

git checkout -b <nom_de_la_branche>

- créer une branche et on switch dans la nouvelle branche.

git checkout <nom_de_la_branche>

- switcher vers une autre branche (deja presente).

git merge <branche_bis>

- fusionner branche_bis avec la branche master.

git branch -d <branche_bis>

- supprimer la branche_bis avec lequel on a fini de travailler.

git branch -D <branche_bis>

- supprimer la branche_bis quand cela ne sert a rien meme s'il n'a pas été fusionner.

git branch -v

- connaître les dernier branches avec des infos en plus sur le dernier commit.

Branche

règle(s): si on a fait des modification dans le fichier d'une branche sans faire de comit, si on revient au branche master cette modification est automatiquement transporter. Il est donc necessaire de faire un commit dans chaque branche avant de se rediriger vers une autre branche quand une modification a ete apporter.

git log --oneline

- afficher un ensemble de comit

git log --graph

- représenter de maniere graphique l'ensemble des comit.

git log -oneline --graph

- fusion des deux commandes

Conflit entre Merge

Cela arrive quand deux personnes effectuent des modifs sur deux branche sur le même travail et qu'il en fait un commit, lors de la commande "git merge <nom_branche>" il y a conflit.

- il suffit de résoudre le conflit en allant directement dans le fichier et de garder la vrai version a suivre.

Alias

Afin de pouvoir raccourcir les commande on cree des alias. Les alias se trouve dans le fichier .gitconfig

git config --global alias.<nom> <commande>

- ajouter une alias.

git <nom_alias>

- executer l'alias

git config --global --unset alias.<nom_alias>

- supprimer l'alias