## Git Mémento

## configuration du compte

|  |  |
| --- | --- |
| git config --global user.name "votre nom ou pseudo" | Configure le nom de l'user |
| git config --global user.email "votre@email.com" | Configure le mail de l'user |ou| user.mail |

## commit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Initialiser** | git init | | | Active un dossier comme repository Git |
| **Ajouter fichier à l'index Git** | git add monfichier.extension | | | Ajoute le fichier a l'index Git |
| git add | | | Ajoute tout les fichiers dans l'index Git |
| **Enregistrer commit** | git commit -m "mon commentaire ici" | | | Enregistre les modification du commit // -m = commentaire |
| git commit -a -m "mon commentaire ici" | | | Mettre a jour un fichier déjà ajouté a l'index = -a |
| **Historique** | git log | | | Affiche l'historique des commit (plus récent en haut) |
| **Naviguer dans l'historique** | git checkout ShaDuCommit | | | Se positionne sur un commit donnée de l'historique |
| git checkout master | | | Revenir a la branche principale (commit le plus recent) |
| **Status** | git status | | | Affiche le status l'etat des fichier (non enregistrer etc.) |
| **Modification** | git commit --amend | | | Permet de refaire le dernier commit (fichier, commentaire) |
| **Retrouver qui a modifier un fichier** | | git blame nomdufichier.exension | | Listes toute les modifications ligne par ligne (avec auteur et sha) |
| git show debutDuSha | | Saisir le début de sha pour retrouver le commit |
| **Ignorer des fichiers** | | Touch .gitignore | Fichier dans lequel ont renseigne les chemin complet des fichier à ignorer(indexé le) | |
| **Conserver temp** | | git stash | Mettre de coté des modifications sans faire de commit | |
| **Restaurer temp** | | git stash pop | Récupérer les modification temporaire sauvegardé | |

## Récupération et envois de code sur le remote GitHub (origin)

|  |  |  |  |
| --- | --- | --- | --- |
| **Récupération** | git clone lienFournieHttps | | Récupère l'intégralité du code d'un repository en https // (ou ssh) |
| git pull origin master | Recup en local les dernières modifications de la branche master de mon remote origin | |
| **Envois** | git push origin master | | Envoie les modifications local dans la branche master de mon remote origin. |

## Branche

|  |  |  |
| --- | --- | --- |
| **Afficher** | git branch | Affiche les branche (celle en cours : \* devant) |
| **Créer** | git branch newBranche | Crée une branche newBranche |
| **switcher** | git checkout master | Se positionne sur la branche master (ajouter -b pour crée **et** se positionné) |
| **fusionner** | git merge brancheB | Fusionne la branch actuel avec la brancheB |
| **Si une fusion échoue pour cause de conflit : modifier les fichiers PUIS faire git commit pour valider le tout (éditer le commit)** | | |
| **supprimer** | git branch -d ma-branche | Supprime la branche ma-branche |

## Remote

|  |  |  |
| --- | --- | --- |
| **Configuration remote** | git remote origin url | Ajoute le remote origin, qui as l'adresse : url |
| git remote -v | Affiche les remotes |