1. afficher l’url de remote origin :

**git remote show origin**

1. git add . : ajouter tous les fichiers modifiés à l’index sauf les fichiers supprimés

git add “\*.html” git add –all ajout les fichiers supprimés

1. Annuler les changements dans le work directory (supprimer modification non indexe)

**git checkout fichier**

1. Annuler les modifications indexes du fichier(le contraire de git add file)

**git reset HEAD fichier** ou bien **git reset -- fichier**

**git reset** : va revenir à l’état unstaged (non indexé) et sauvegarde les modifications (dangereux on ne peut pas la défaire)

**git reset --hard** : va revenir à l’état unstaged (non indexé) et a les modifications

**git reset commit\_id :** supprimer toutes les commites postérieurs à « commit\_id » mais garde l’espace de travail intacte.

**git reset HEAD^** : retourne d’un cran (un seule commit)

**git reset HEAD^^** : retourne de deux crans (2 commits)

**git reset HEAD^** **(soft, mixed, hard) = (modifications stagé on fait Just un commit, modifications non stagé on fait add et un commit ,supprime les modifications)**

Cette commande n’est pas sécurisé elle retourne en arrière et supprime l’historique

1. Afficher les détails des modifications non indexes avec la versions dans le dernier commit

git diff

1. git log don l’historique des comites

* git log –n 2 : don les 2 dernières commites
* git log –oneline : on une seule ligne s’il yà bcp de commits
* git log –p path : voir les commits spécifique à un fichier ou un répertoire

1. git commit –a –m pour abrégé (-a pour stagé)
2. git checkout « id du commit » : navigué dans l’historique des commit (voir le fichier comment il est avant ce commit)

**git checkout maste**r : retourné à la branche mater

**git checkout idcommit filename :** annuler les modification sur le fichier filename(défaire ce qu’on fait)

1. git revert commit\_id : défit un commit (supprimer les changements de ce commit) revert et ajouté à l’historique
2. git stash
3. git branch : pour tester une expérimentation (changer le code source sans être sur que je vais le sauvardé)
4. git checkout –b test : créer une nouvelle branche et se met sur la nouvelle branche teste
5. git checkout nom de la branche
6. conflits
7. logging avec blame
8. git show sha : voir exactement ce qui est passé dans le commit
9. ignorer des fichiers qu’on veut pas envoyé au repot distant mais resterons localement en ajoute un fichier «.gitignore » puis ajouté sur chaque ligne les fichiers et les dossiers à ignoré
10. git stash suspendre un travail puis le reprendre après.
11. Git stash pop recharger le stash et supprime ce qu’il y’à dans le stash
12. Config

Soit local propre au dépôt au global pour tous les dépôts. Les deux variable à configurer et user.email et user.name ..etc

Pour voir la configuration on fait git config –list

1. Git status : l’état du dossier
2. Zone stagging zone de transit

IdblobFichiernom1

|  |  |
| --- | --- |
| Fichiernom1 | IdblobFichiernom1 |
| Fichiernom2 | IdblobFichiernom2 |
| … | …. |

Répertoire de travail

Fichiernom11

Fichiernom21