GIT

Git config –global variable : configure préférence git

Git config –list : liste les variables de configuration

Git init : initialise un repository GIT

Git remote add + nom du projet + adresse git : pointe le project sur un repository github

Git clone + adresse du repository : copie l’ensemble des fichiers du repository

Git branch : liste les branches du repository

Git branch + nom d’une branche : crée une nouvelle branche

* d + nom d’une branche : supprime la branche
* D + nom d’une branche : force la suppression de la branche

Git Fetch : récupère les donnés de github sans fusionné avec la branche principale

Git merge + branche : fusionne la branche avec la branche pointé.

Git pull : réceptionne les modifications en local depuis github en fusionnant avec la branche principale = Git fetch + git merge

Git checkout + nom d’une branche : se positionne sur la branche

* b + nom d’une branche : crée la branche et se positionne dessus.

Git status : permet de voir les fichiers indexés et modifiés

Git add + fichier : index les fichiers à commiter

Git commit -m + description du commit : effectue un commit

Git commit –amend -m + description du commit : modifie la description du dernier commit

Git commit –amend –no-edit : ajoute les fichiers indéxé via git + add + fichier dans le dernier commit

Git reflog ou git log : liste la liste des commits

Git blame + non du fichier : affiche les personnes qui ont modifié le fichier.

Git push : envoie les modifications sur github

Git stash : permet de mettre de côtés des mofidications.

Git stash list : liste les stashs

Git stash apply + stash@{x} : rapatrie les modifications du stash

Git reset –hard HEAD^ : supprime le dernier commit de la branche

Git reset –hard + sha du commit : supprime le commit

Git reset –soft + sha du commit : permet de se positionner sur un ancien commit avec les donnée de l’ancien commit.

Git reset –mixed HEAD^ : revien juste apres le dernier commit sans supprimé les modifications, crée un head détaché, il désindexe les fichiers.

Git reset –mixed + sha du commit : supprime le commit

Git reset va par défaut effectuer un Git reset –mixed HEAD^

Git revert HEAD^ : annule un commit public en cas d’erreur en créant un nouveau commit.

Git cherry-pick + sha : dupplique un commit d’une autre branch dans la branche pointé.

Git rebase : permet la refonte des commites pour restructurer et éclaircir l’historique de codage du projet.

* # p, pick = utilisez le commit
* # r, reword = utilisez le commit, mais éditez le message de commit
* # e, edit = utilisez le commit, mais arrêtez-vous pour apporter des changements
* # s, squash = utilisez le commit, mais intégrez-le au commit précédent
* # f, fixup = commande similaire à "squash", mais qui permet d'annuler le message de log de ce commit
* # x, exec = exécutez la commande (le reste de la ligne) à l'aide de Shell
* # d, drop = supprimez le commit

Ssh-keygen -t rsa -b 4096 -C « johndoe@example.com » : crée un clée public et privée