Formation Pratique Git et Github

Git : est un système de contrôle de version, qui nous permet de gérer les différentes versions de nos projets

Github : c'est une plate forme qui nous permet de conserver les différentes versions de nos projets

Avant toute chose, il faut d'abord commencer par installer le logiciel Git dans son ordinateur. Après son installation, Git nous propose une interface console (GitBash) et une interface graphique (GitGUI).

Quelques Commandes Git

git –version : pour vérifier la version du Git qui est installé

git config

git config –global user.name « saisie de votre nom » : permet de configurer votre nom

git config –global user.email « saisie de votre email » : permet de configurer votre adresse E-mail

NB. A chaque fois que vous voulez changer votre nom ou votre email, vous allez toujours tapez les mêmes commandes

git config –global user.name : permet de d'afficher le nom que vous avez configuré dans git

git config –global user.email : permet de d'afficher l'E-mail que vous avez configuré dans git

mkdir chemin du dossier (ex : C:/Dossier1/Nouveau Dossier) : permet de créer un nouveau dossier à partir de git

mv C:/DossierParent1/Dossier1 C:/DossierParent2/ Dossier1 : permet de déplacer le Dossier1 se trouvant dans DossierParent1 vers le DossierParent2

cd chemin du dossier (ex : C:/Dossier1/Nouveau Dossier) : permet d'accéder à un dossier

ls : permet d'afficher les contenus d'un dossier. On le fait après avoir accéder au dossier

ctrl + l : permet de nettoyer le consol git

git init : nous permet de faire un repository ou dépôt c'est-à-dire initialiser un projet entant qu'un dépôt git. On le fait après avoir accéder au dossier du projet

git status : permet de vérifier les statuts du contenu du dossier du projet, pour savoir s'il y a eu des modifications ou pas

git add nomfichier.extention : permet à git de prendre en compte ou de sauvegarder les modifications apportées au fichier dont le nom et l'extension a été mentionné précédemment

git add –A : permet à git de prendre en compte ou de sauvegarder les modifications apportées à tous les fichiers du dossier

git commit –m « nom du backup » : permet de générer le backup de toutes les modifications qui ont été effectuées. Cette commande est tapée juste après la commande git add nomfichier.extension ou git add -A

git log : C'est pour visualiser tous les commits ou tous les backups générés

touch fichier.extension : c'est pour créer un fichier

git diff fichier.extension : permet de visualiser les modifications apportées au fichier. On tape cette commande lorsqu'on n'a pas encore fait git add et git commit

git reset - -hard ID commit : permet de revenir au commit dont l'ID est spécifié, puis annuler ou supprimer le commit qui vient après en annulant aussi les modifications qui ont apportées au fichier

git reset - -soft ID commit : permet de revenir au commit dont l'ID est spécifié, puis annuler ou supprimer le commit qui vient après sans annuler les modifications qui ont apportées au fichier

git commit - -amend - -no-edit : permet d'ajouter la sauvegarde dans le commit qui a été généré en dernier lieu sans pour autant créer un nouveau commit

git show ID Commit : permet de visualiser les modifications qui ont été apportées au commit dont l'ID est spécifié

git branch : permet de lister toutes les branches

git checkout –b nom de la branche : permet de créer une nouvelle branche et de basculer ou de switcher dans celle-ci

git checkout nom de la branche : permet tout simplement de basculer ou de switcher dans une branche

git merge nom de la branche : permet de fusionner la branche avec la branche master. Pour faire git merge, il faut d'abord faire un commit sur la branche créée, puis sur la branche master (facultative) en suite faire git merge nom branche qui a été créée. git merge se fait toujours dans la branche master

git branch - -d nom de branche : permet de supprimer une branche lorsqu'elle a déjà été fusionnée avec la branche master

git branch - -D nom de branche : permet de supprimer une branche lorsqu'elle n' a pas encore été fusionnée avec la branche master

git config - - global alias.s status : permet de créer un alias pour la commande status. C'est idem pour d'autres commandes

git config - -global - -unset alias.s : permet de supprimer l'alias s qui a été crée

git log - - oneline : permet pour resumer l'affichage de commit (on le fait quand nous avons plusieurs commit)

git log - - graph : permet de lister le commit sous forme graphique

git log - - oneline - - graph : c'est pour exécuter les commandes précédentes

git remote –v : permet de lister tous les dépôts distants

git push -u origin master(main) : permet de faire un push ou un dépôt à distances (sur github). On le fait lorsqu'on dépose pour la première fois un projet. Et après nous commencer à faire tout simplement : git push

NB. Gerating ssh keys windows : c'est une procédure qui va nous permet de sécuriser les dépôts sur git hub