GIT

Git est un outil de versioning de code (logiciel de gestion de version), c’est-à-dire le faite de garder des traces (commit) sur toutes les modifications que vous avez fait dans votre code

Un commit est correspond à une version de votre code dans un instant t

On a principalement deux modèles de logiciels de gestion de version :

* Modelé centralisé : c’est-à-dire qu’on a un serveur central qui contrôle toute la base de code du logiciel, dans ce cas, on a toujours besoin d’une connexion internet pour qu’on puisse se connecter à ce serveur
* Modèle distribué : toutes les machines ont accès à la base de code, c’est-à-dire que chaque développeur à l’intégralité de tout le code dans sa machine en local, donc, ce cas, on n’a pas besoin d’un serveur central, donc, pas besoin d’une connexion internet.

Console (terminal) dans lequel on exécuté des commandes (cd, vim, etc.)

Voici quelques commandes de base :

* pwd : pour connaitre votre répertoire courant
* ls : pour lister l’ensemble de fichiers et répertoires de dossier courant (il ya également des options pour afficher plus d’informations comme « l » et « a ». exemple : ls –l -a
* cd : pour se déplacer dans un répertoire, exemple :
  + cd [nom\_dossier]
  + cd .. pour revenir au répertoire parent
  + cd ~ pour revenir dans votre répertoire principal
* touch : pour créer un fichier (touch file.txt)
* mkdir : pour créer un dossier (mkdir dossier)
* cat : pour afficher le contenu d’un fichier (cat file.txt)

Installation GIT

Une fois « git » est installé, il faut configurer votre compte git pour créer ensuite un compte github

git config - -global user.name ‘’name ou pseudo’’

git config - -global user.email ‘’mail’’

Pour lister es réglages de git utilisés, taper la commande : git config - -list

RQ : pour coller du texte dans git bash, il faut utiliser le boutton « insert » de votre clavier

* Vim est un éditeur de texte très performant, pour ouvrir un fichier avec vim, il faut utiliser la commande « vim file.txt »
* Commandes éxécutables depuis vim
  + w : pour sauvegarder le fichier
  + q : pour quitter vim
  + x : pour enregistrer et quitter vim

Faire votre premier commit

* Pour activer un dossier comme repository git, il faut d’abord se déplacer dans ce dossier, et lancer la commande « git init », après, git génère un fichier index « .git » qui contient tous les fichiers dont il a besoin pour faire suivre ces fichiers
* Une fois que vous créez des nouvelles fichiers dans ton repository, il faut les ajouter dans l’’index en lançant la commande « git add [file.extention] » pour que git pour suivre les modifications qui seront faite dans ce fichier
* Pour gagner du temps, pour pouvez ajouter les fichiers dans le répertoire courant en tapant la commande « git add . »
* A chaque fois que je fais une modification, vous devez demander à git d’enregistrer vos modifications, pour faire ça, on lance la commande « git commit –m ’’un message qui spécifie la modification que vous avez fait’’ »
* « git status » pour savoir où on est dans les modifications de notre projet (c’est-à-dire s’il Ya des commit à faire par exemple)
* Pour afficher l’historique des commit, il faut lancer la commande « git log » dans le répertoire de notre projet, on part du commit le plus récent au commit le plus vieux

Dans la liste de cet historique, chaque commit est répertorié avec :

* + son **SHA** : son identifiant unique, qui se présente sous forme d'une longue chaîne de caractères et de nombres. Par exemple : "87753191cef0bdb955a4cb4ff841f7c2cce4cb1c".
  + son auteur : qui a fait le commit (utile lorsque vous travaillez à plusieurs sur un projet !)
  + sa date
  + sa description : vous vous rappelez, c'est le message de description que vous indiquez avec l'option  -m lorsque vous faites votre git commit
* à chaque fois qu’on modifie un fichier, il ya deux étapes à faire :
  + ajouter le fichier dan l’index « git add . »
  + faire le comit
* On peut condenser ces 2 étapes (add and commit) en une seule, si on mit à jour les fichiers qui existent déjà dans l’index : « git commit –a –m ‘’message’’ ». l’option « a » demande à git de mettre à jour les fichiers déjà existant dans son index

Positionnez vous sur un commit donnée

* Pour se positionner sur un commit donné, il suffit de lancer la commande « git checkout [SHADucommit] »
* Pour retourner à la branche principale, on lace la commande « git checkout master »
* En git, on ne peut pas vraiment supprimer un « commit », mais on peut l’annuler à travers quelques options, mais il faut utiliser ces options avec prudence. Donc on peut « revert » un commit, c’est-à-dire créer un nouveau commit qui fait l’inverse du précédent, on fait ça avec la commande « git revert SHADucommit », ça crée un nouveau commit dans l’historique du coup
* On peut également modifier le message du dernier « commit », mais on peut faire ça seulement avant de faire le push de notre commit sur l’origine. On utilise la commande « git commit - -amend –m « nouveau message »
* Parfois, on fait un certain nombre de changements dans mon code source, mais on veut annuler tous les changements que je n’ai pas encore comité, pour faire ça on utilise la commande « git reset - -hard »