# Le commit :

Le commit permet de « versionner » son code. Un commit est suivi d’un message indiquant pourquoi on commit. En local, un commit est une version du code. On peut le mettre à jour avec un nouveau commit ou revenir à une ancienne version grâce à la liste des commit. En réseau, c’est le même procédé mais on synchronise son travail en ligne pour ensuite pouvoir travailler sur n’importe quel PC équipé de Git. Grâce à un commit, plusieurs personnes peuvent travailler sur un même projets en même temps via plusieurs branches ou à des heures différentes sur la même branche. Le commit met à jour certain fichier en supprimant ou ajoutant les éléments en différence avec la version actuelle.

Pour commit :

* « git add . » pour ajouter tous les fichiers
* git commit –m « message »
* git push (si on synchronise en ligne)

# Git log

Git log permet de voir qui a modifié quoi. On peut avoir une liste comprenant le pseudo, le nom du commit, la date, le fichier et ce qu’il a modifié dans le fichier. C’est un outil important dans la compréhension des changements du code.

Utilisation :

* « git log » pour avoir la liste.

# Une branche :

Une branche est une section sur laquelle on peut synchroniser du travail. La branche principale étant la branche Master.

Si plusieurs personnes commit en même temps sur une même branche il y aura un conflit. En effet on aura deux modifications et on ne pourra pas savoir lequel est prioritaire. Autre cas, si on veut d’abords terminer entièrement une étape avant de commit, on peut la terminer sur une branche en dehors de la branche principale pour la commit seulement quand celle-ci sera terminé, ou après vérification par l’administrateur.

L’objectif de créer une nouvelle branche est de créer une branche parallèle à la branche master pendant un certain temps pour ensuite amener un code complet sur la branche master.

Utilisation :

* « git branch nouvelle »
* « git checkout nouvelle » pour accéder à cette branche et la modifier.

Pour fusionner des branches, par exemple pour fusionner la branche nouvelle dans la branche Master, on se place sur la branche master puis on ramène la branche nouvelle dessus :

* « git checkout master »
* « git merge nouvelle »

A présent les branches sont synchronisées sur la branche master.