

Git et GitHub (Partie 3)

Note : Je sais qu'il ne faut pas utiliser des « et des », c'est juste word qui fait la merde ^^

Qu'est ce qu'un commit ?

Un commit est une sorte d'entrée de journal qui sert à indiquer les fichiers qui ont été modifiés, ajoutés ou supprimés sur un projet. Il se fait avec la commande suivante :

git commit -m « message »

L'avantage de ce genre de système est qu'un développeur lambda qui rejoint le projet peut facilement voir son historique. De la même façon, on peut, en cas d'erreur, revenir sur le commit précédent. Il faut veiller à ce que le message accompagnant le commit, soit le plus clair possible pour que les personnes qui seront amenées à travailler sur le projet.

La commande git log

La commande git log sert à afficher le journal du repo, il contient la liste de toutes les actions faites sur celui-ci. Il peut s'agir aussi bien de commit (voir ci-dessus), que de création de branches, ou en de *merge* (fusion de branche).

Pour quitter le journal, on appuie sur la touche Q.

Qu'est ce qu'une branche ?

Une branche est une bifurcation du projet principal qui peut par exemple être utilisée pour tester une nouvelle fonctionnalité. Par la suite, si on le désire, on peut utiliser la commande *git merge* pour fusionner une branche avec la branche principale (traditionnellement master).

Une branche est créée avec la commande suivante :

git branch masuperbranche

Si on veut on peut aussi afficher la liste des branches, git indique la branche sur laquelle est travaillé avec une *

git branch

Ou encore changer de branche de travail :

git checkout masuperbranche