Git & GitHub  
Activité Partie 3

Table des matières

[Qu’est-ce qu’un commit : 1](#_Toc12194635)

[À quoi sert la commande git log : 1](#_Toc12194636)

[Qu’est-ce qu’une branche : 1](#_Toc12194637)

# Qu’est-ce qu’un commit :

Pour simplifier la chose, dites-vous qu’un commit est un peu comme faire une sauvegarde d’un fichier Word, Excel, txt,… ou tout autre document pour lequel vous auriez fait un simple CTRL-s.

« *Soit, mais alors pourquoi s’embêter avec GIT, si ce n’est rien de plus que ça ?* »

Et bien la grande différence réside dans le principe du versionning, c’est-à-dire, de conserver une copie des anciennes sauvegarde, afin de pouvoir y accéder de nouveau si nécessaire.

Contrairement à des logiciel de traitement de texte comme Word, lorsque vous faite une sauvegarde, l’ancienne version de votre fichier est écrasé par la nouvelle, sans aucun moyen de pouvoir y accéder de nouveau, à moins de faire une copie à part entière de votre document à chaque sauvegarde, ce qui pourrai devenir compliqué à gérer, si vous deviez créer un document conséquent demandant un grand nombre de mise à jour, de travail et donc de sauvegarde.

En bref, un commit est une sauvegarde, ou, contrairement à d’autre logiciel simple, GIT permet de conserver une trace de chacune d’elle, vous donnant également la possibilité de pouvoir y naviguer et ainsi de retrouver l’état de vos fichier à l’instant T du commit.

Voici la commande « git commit -m "monCommentaire" »

# À quoi sert la commande git log :

Vous vous rappelez de ce que je vous ai expliqué pour le commit ?  
Et bien la commande « git log » permet justement de visualiser la liste de vos commit.

« *OK, bon, imaginons que j’utilise GIT et fais des commit pour le suivi de mes sauvegarde, mais que j’en ai des dizaines et des dizaines, je fais comment pour m’y retrouver dans cette liste?* »

Et ben c’est simple, au moment de faire votre commit

# Qu’est-ce qu’une branche :