**Git & GitHub - Activité N°3**

1. Qu’est-ce qu’un commit :

Le commit sert à envoyer et valider localement les changements apportés à un projet et rajoute ces informations dans l’historique des actions réalisées.

On s’en sert principalement pour valider une étape de son projet (rajout de code, modification de la documentation, etc…) .

Il s’utilise avec la syntaxe :

git **commit** -- Commande de base

git **commit** **-**m "Voici mon premier commit"

-- Commande avec le message associé au commit

git **commit** **-**am /\* Équivalent à git commit -a -m "message\_du\_commit"

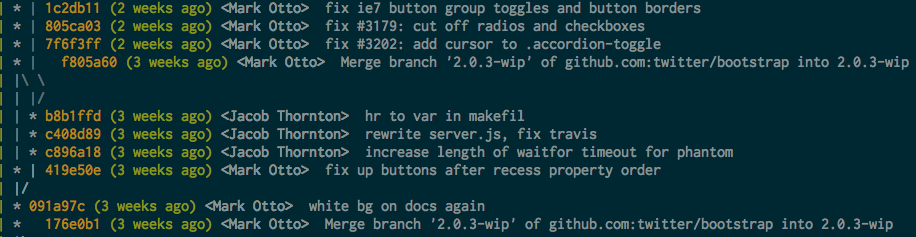
/\* L’option –a applique les changements à un fichier présent

/\* dans l'index qui a subi des modifications. \*/

On évite générallement de faire des commits si un travail en particulier n’a pas été complété (afin de clarifier un maximum les actions réalisées) et on définit un message de commit le plus explicite possible.

1. A quoi sert la commande « git log » :

Cette commande sert à lister la liste des différents commits apportés à un projet. Les informations données sont le n° unique (sha), le nom de l’auteur, la date et le message associé aux commits existants.



1. Qu’est-ce qu’une branche ?

Une branche sert à développer un nouveau besoin ou une nouvelle fonctionnalité à un projet de façon séparée. On distingue la branche principale (master) des autres branches.

git **branch** –- Affiche la branche en cours

git **branch -v** –- Visualiser les derniers commits sur chaque branches

git **branch** **ma\_branche** –- Crée une branche nommée « ma\_branche »

git **checkout** **ma\_branche** -- Permet de se déplacer sur « ma\_branche »

git **checkout** **-**b **ma\_branche** /\* Combine la création et le déplacement

/\* sur « ma\_branche » \*/

git **branch** **-**d **ma\_branche** /\* Supprime « ma\_branche »