

# Explications des concepts de Git

## 1.Qu'est-ce qu'un commit :

Faire un commit, c'est faire une sauvegarde des modifications de son développement en cours. Lorsqu'on estime que notre code mérite d'être sauvegardé, on lance donc la commande « `git commit -m "nom-de-la-modification"` », et une sauvegarde est créée.

Il est très important de donner un nom le plus clair possible au commit, afin de comprendre instantanément ce qu'il contient, et donc de le retrouver rapidement.

L'intérêt de faire des commit, c'est qu'il est possible de revenir à n'importe quel moment sur un commit donné. En somme, Git est un peu une machine à voyager dans le temps, dont les commit seraient les différentes dates à visiter.

Lors d'un commit, Git génère automatiquement un identifiant associé au commit. Cet identifiant permet de retrouver le commit

## 2. A quoi sert la commande git log :

La commande `git log` permet d'afficher un journal des différents commit sous forme de liste, ce qui permet de voir rapidement (à condition que les commentaires des commit soient clairs) quels commit ont été fait et leurs contenus.

On peut donc se repositionner sur un commit donné si l'on souhaite revenir en arrière dans son travail.

## 3. Qu'est-ce qu'une branche :

Créer une branche, c'est faire une copie de l'historique principal des commit (appelé branche « `master` »), pour travailler dessus sans affecter ladite branche principale.

Cela permet donc de casser le code, le modifier ou le saccager sans retenue sans pour autant perturber le développement de la branche `master`.

C'est idéal si l'on souhaite essayer de nouvelles choses, sans être sûr du résultat. Si le résultat fonctionne, on peut alors fusionner la branche secondaire avec la branche `master` pour intégrer notre travail au développement principal du code.

Le schéma ci-dessous permet de clarifier ce concept :

