

Partie 3 Activité

1. Qu'est-ce qu'un commit ?

Le développement Web, et le code en général nécessite beaucoup d'allers-retours et de modifications. Il est nécessaire d'avoir une trace des différentes évolutions pour, notamment corriger les bugs. De plus, travailler sur du code en collaboration va demander de la rigueur et de l'organisation. Fonctionner avec un logiciel de versionning va satisfaire les exigences du travail collaboratif sur du code.

On appelle commit, l'enregistrement de la version de fichiers, indexés au préalable, présent dans le dossier de travail ou repository. Il sera désormais possible d'avoir accès à une version donnée après plusieurs commit. Cela va permettre d'avancer son travail en enregistrant des points de repères, pour lesquels on va pouvoir identifier en quoi consiste le commit grâce à un message avec la commande suivante :

`git commit -m « messages court donnant une indication sur le commit »`

Ainsi, si la direction que prend le projet n'est pas satisfaisante, il est possible de revenir à un commit plus fiable et repartir dans la direction souhaitée.

2. A quoi sert la commande git log ?

La commande git log donne accès à l'historique des commit, et permet de voir à quel moment et qui a effectué un commit et en quoi consiste le commit si un message a été laissé par l'auteur. Il permet aussi d'obtenir le SHA du commit, c'est à dire l'identifiant qui va permettre de revenir à la version du commit avec la commande suivante :

`git checkout SHA`

3. Qu'est-ce qu'une branche ?

Une branche permet de prendre une direction parallèle pour, par exemple, tester de nouvelles fonctionnalités. L'intérêt réside dans le fait que les fichiers de la branche master ne seront pas affectés par les commit réalisés sur les autres branches.

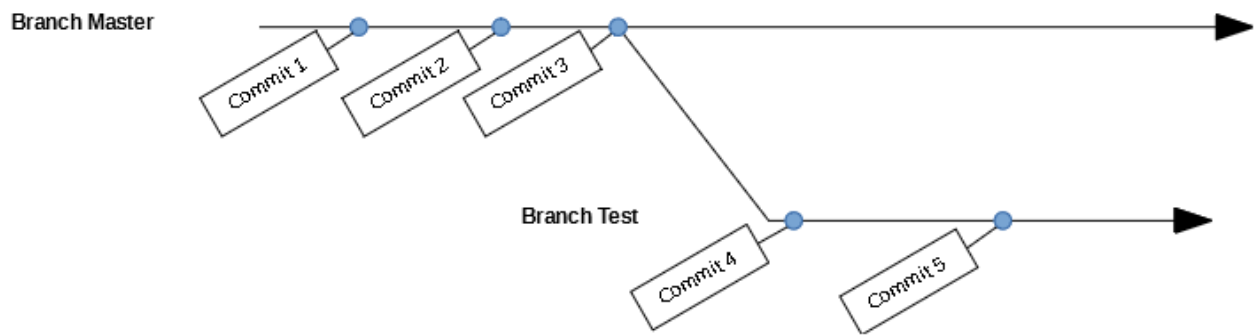


Illustration 1: Les branches

De plus, si jamais les fonctionnalités testées sur une branche fonctionnent et sont satisfaisantes, il est alors possible de fusionner le dernier commit de la branche avec celui de la branche master grâce à la commande suivante :
`git merge nomDeLaBranche` (manip à réaliser dans la branche master)

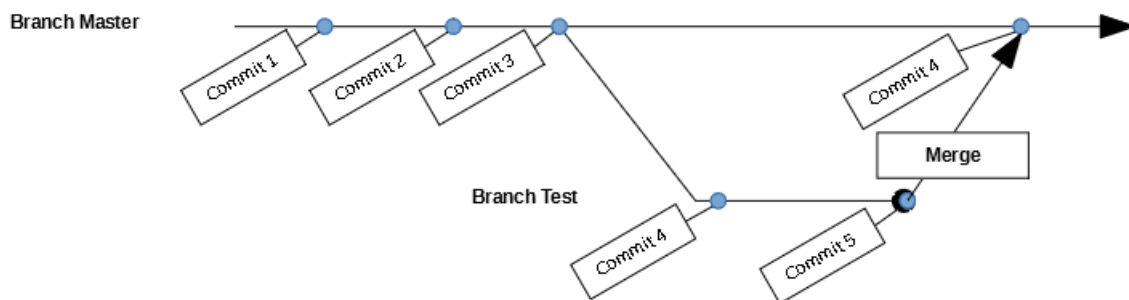


Illustration 2: Merge d'un commit d'une branche sur la branche master