**Utilisation de Git et GitHub**

**Présentation :**

Git est un logiciel de gestion de versions décentralisé open source. Il permet de stocker un code informatique pas uniquement en local sur l'ordinateur de l'utilisateur, celui-ci peut aussi être stocké sur un serveur dédié.

Github quant à lui est une plateforme d'hébergement de projet / de code open source elle aussi. Il permet de faciliter la collaboration autour d'un projet entre différents utilisateurs qui sont alors en mesure de le récupérer, dupliquer ou de proposer des modifications sur du code si le propriétaire du projet accepte.

**Quels sont les utilités d'un système de gestion de version ?**

L'idée est d'utilisé un serveur distant qui contiendrait un historique de toute les modifications réalisées par chaque contributeur d'un projet et ainsi mettre en commun les avancées de chacun. Le système distribué que propose Git et Github permet à chaque utilisateur d'héberger l'entièreté du code du projet sur lequel il travail sans avoir besoin d'être constamment connecté au serveur distant. De plus chaque collaborateur ayant une version propre du projet celles-ci peuvent être utilisé en tant que backup dans le cas d'une corruption totale du projet par un autre collaborateurs.

**Gestion d'un projet grâce à github :**

Il est nécessaire d'avoir git installer sur notre ordinateur comme prérequis.

Dans un premier temps nous allons nous intéresser au fonctionnement de git à l'aide de ligne de commande nous verrons par la suite qu'il existe d'autres pour se faire.

Principe de fonctionnement de git :

Les données dans git sont assimilés en réalité à un flux de snapshots, à chaque validation/enregistrement d'une session de travail sur notre projet git, celui-ci va créer un snapshot de notre espace de travail et en créer une référence afin de nous y donner accès par la suite. Ce snapshot est stocké sur notre ordinateur en local dans une base de données.

Les fichiers dans git on plusieurs états possibles dont 2 grand états à savoir : suivi (fichier enregistré sur la base de de donnée) et non suivi (non enregistré sur la base de données et donc n'appartient pas au dernier snapshot).

Chaque fichier suivi peut lui-même avoir un autre sous état :

* Modifié (modified) : lorsque le contenue du fichier et modifié comparé à celui du dernier snapshot.
* Indexé (staged) : lorsque que l'on indique à git que le fichier fera partie du prochain snapshot/
* Validé (commited) : lorsque les fichiers du snapshots sont stockés dans la base de données locale.

Chaque projet Git est composé de 3 sections distinctes :

* Le répertoire de travail (Working Tree) : Correspond au dossier qui contient réellement nos fichiers extrait de la base de données sur laquelle sont stocké nos snapshots afin de pouvoir les modifiés et/ou les utiliser.
* La zone d'index (Staging Area) : Fichier qui stocke le contenu du prochain snapshot.
* Le répertoire Git : Stocke la base de données de notre projet.

Les branches permettent de développer des fonctionnalités isolées les unes des autres. Par défaut la branche principale est la master, on en crée d'autre pour le développement que l'on fusionne ensuite à la branche principale une fois terminé.