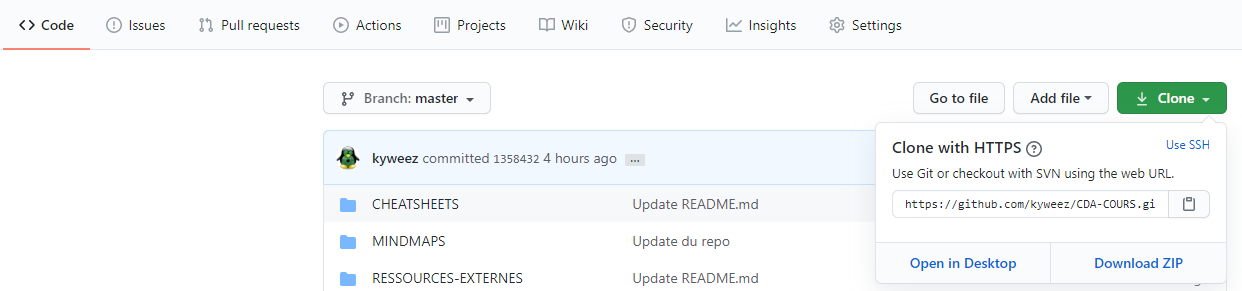


|  |
| --- |
| Gestion de branche avec flow |
|  |
| 26 juin  Concepteur Développeur D'Application  Créé par : Jonathan Jeanniard |

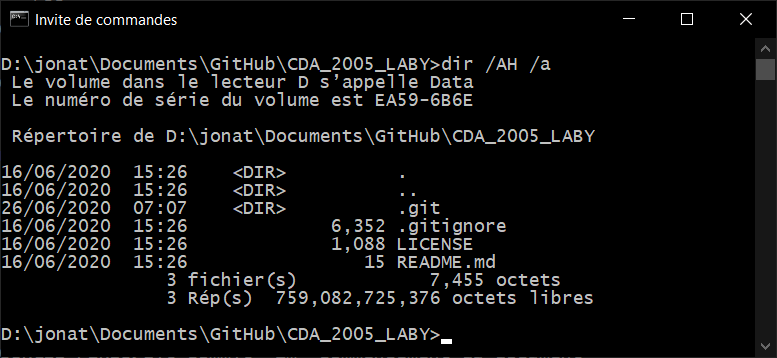
1. Récupération du dépôt

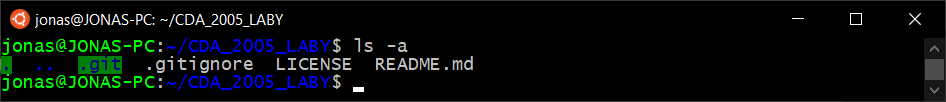
Dans un premier temps récupérer un dépôt quel qu'il soit avec git clone.



1. Vérification git

Ouvrer une invite de commande(cmd), pour Windows et la console (Terminal), pour Linux. Navigué jusqu'au répertoire où se trouvent le dépôt cloner avec la commande "cd" pour les deux systèmes.

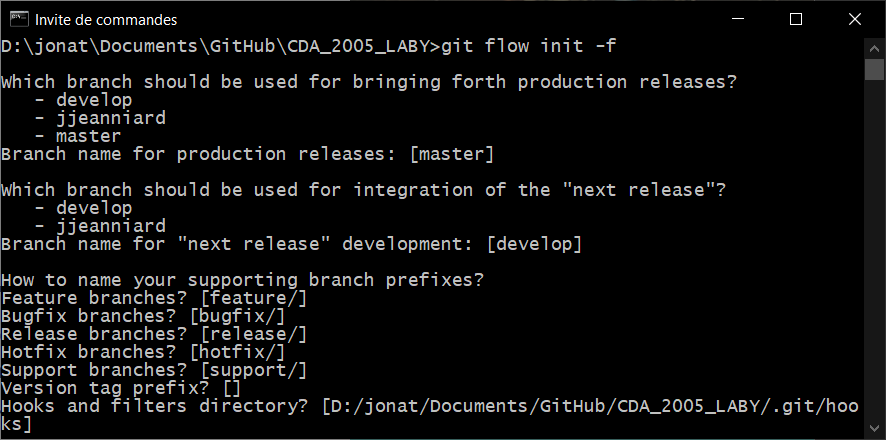




Une fois dans le répertoire du dépôt vérifier, que se trouve le dossier cache de git avec la commande "dir /AH /a" pour Windows et pour Linux "ls -a".

1. Initialisation de flow

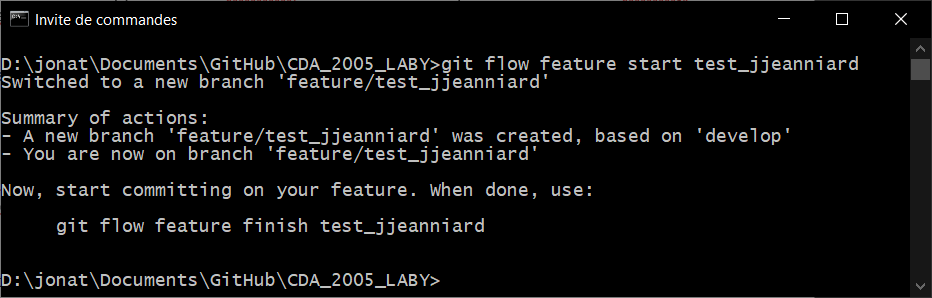
A partir de là, les commandes de git sont communes au 2 systèmes, je vais utiliser le cmd de Windows pour la suite et pour plus de facilité. Donc pour initialiser la gestion des branches, on tape "git flow init".



De nombreuse questions vous sont posées, vous pouvez tout mettre par default en appuient sur entrée (si un problème survient au moment de "next release", il faut au préalable créer la branche "develop"). Chaque branche corresponde une fonction donnée, comme vous pouvez le remarquer.

1. Première branche (pas le vieux)

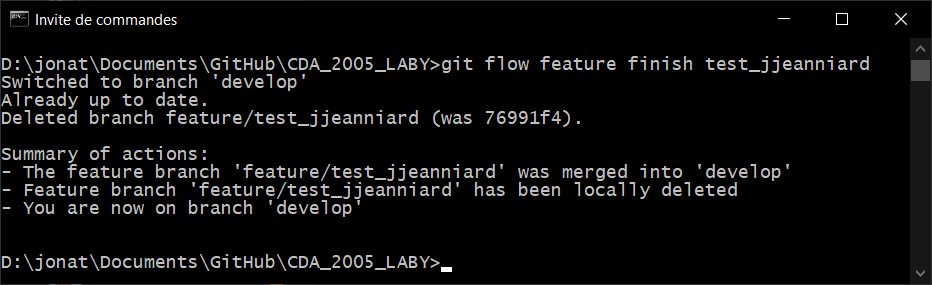
Ma première fonctionnalité de mon programme sera "test", donc la commande pour réaliser mon espace de travail est "git flow feature start test\_jjeanniard". J'ai par convention de mettre mon pseudo ou nom, précédé d'un underscore après le nom de la fonctionnalité. Cette commande va créer une nouvelle branche tout en récupérant ce qui se trouve dans la branche "develop" et puis vous change automatiquement de branche en vous placent dans la nouvelle.



Ps : Dans les livres, ils préconisent de commencer dans la branche "feature" avec pour nom "init\_projet" pour la réalisation d'un nouveau projet.

1. Fin de la branche

Après de nombreux commit, push, du "code review" sur la fonctionnalité (cycle sans fin) et un "pull request" réussie sur Github. Il est temps de changé de branche, mais avant d'en créer une nouvelle, il est préférable de rendre la liberté à votre précédente branche avec la commande "git flow feature finish test\_jjeanniad".



Un merge de votre branche se fait automatiquement, avant d'être supprimer et puis vous êtes de nouveau switch dans la branche "develop". Une nouvelle vie de branche peu commencer (feature, hotfix, bugfix…).

Une chose à savoir, il n'est pas interdit de travailler sur plusieurs branches. A vous de savoir gérer entre et de ne pas modifier le mauvais fichier avec la mauvaise branche.

Good reading.

Crédit :

Réalisation par : Jonathan Jeanniard, le 26/06/2020 pour la découverte de la commande "git flow".