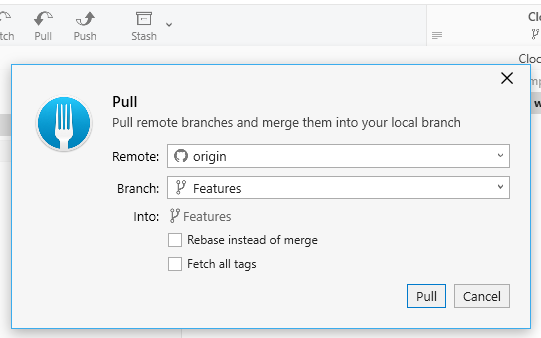
Marche à suivre git

## Pull from github :

Avant chaque changement sur git, il est impératif de récupérer la dernière version sur github pour être sûr de travailler sur le même projet.



Le mieux est d’importer sur la branche la plus récente, dans mon cas *features*, et de merger ensuite vers master (point suivant).

## Merger en local :

Comme on peut considérer la version sur github comme la dernière mise à jour (master), je conseil de merger son master en local vers la version récupérée.

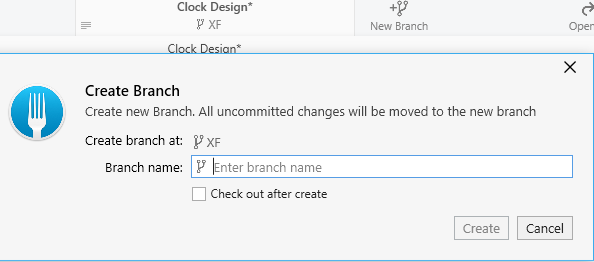
Pour cela, il faut d’abord checkout le master, donc double clic sur master. Ou bien clic-droit sur master -> master -> checkout.

Ensuite clic-droit sur la branche en question, merge into master.

Ne pas oublier de travailler sur une autre branche que master après cela, ou bien d’en créer une nouvelle (point suivant).

## Creating new branch :

Créer une nouvelle branche est essentiel pour assurer qu’on ne change pas la branche master par inadvertance. Dans notre cas, la branche master est protégée est donc impossible à pousser/merger vers l’orgine, cette étape est donc obligatoire !



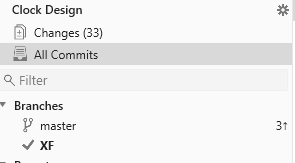
Maintenant, libre à l’utilisateur de travailler toujours sur la même branche crée ou d’en faire une nouvelle à chaque fois.

**Attention :** vérifier qu’on travaille bien sur la branche crée (petite flèche à côté).

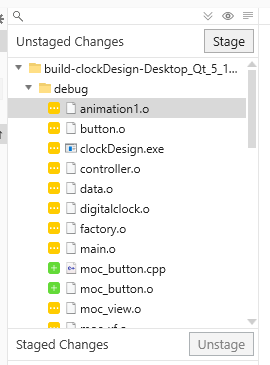
Pour cela, double clic sur la branche sur laquelle on veut travailler.

## Staging files :

Lorsqu’on a modifié des fichiers, on peut voir sur le menu *Changes* le nombre de changement effectuer.



C’est pourquoi il faut d’abord ajouter ces modifications avant chaque commit :



Si on veut seulement ajouter certain fichier, il faut les sélectionner et appuyer sur *stage*. Pour tous les ajouter d’un coup, il faut appuyer sur le bouton en haut à droite (la flèche).

## Commit files :

Après un stage, on peut enfin commit les modifications. Il faut ajouter un titre, description (facultatif) puis bouton commit.



## Push to github :

## Push :