

Rôle des commandes (GIT)

1. Git config

Git config est un outil permettant de voir et modifier les variables de configuration qui contrôlent tous les aspects de l'apparence et du comportement de Git.

On peut par exemple spécifier notre identité grâce aux commandes :

```
$ git config --global user.name "NOM" -> nom de l'utilisateur  
$ git config --global user.email [Adresse mail] -> adresse mail liée à  
toutes les opérations de commit
```

On peut également choisir l'éditeur de texte utilisé

```
$ git config --global core.editor [editeur]
```

Par la suite on pourra vérifier ses paramètres grâce à la commande :

```
$ git config --list
```

2. Git init

Cette commande permet de créer un nouveau dépôt .git vide, mais également des sous-répertoires.

```
$ git init [nom-du-projet] -> Crée un dépôt local à partir du nom du projet
```

3. Git status

Cette commande affiche l'ensemble des nouveaux fichiers, ainsi que les fichiers modifiés à commiter.

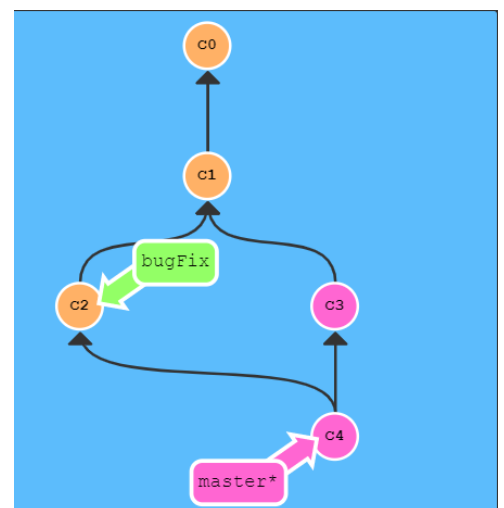
4. Git merge

Git merge fusionne deux branches, cette commande permet

de faire avancer la branche courante de sorte qu'elle incorpore le travail d'une autre branche.

5. Git diff

Git diff permet d'afficher les modifications entre l'arbre de travail et l'index ou un arbre. Cela est utile pour visualiser ce qui a été modifié mais pas encore indexé.



Emilie
JEANNELLE
INFO1

6. Git blame

La commande git blame permet de connaître l'utilisateur qui a fait la dernière modification. Elle permet ainsi de savoir qui a pu engendrer un bug pour pouvoir le corriger.