git_cheatsheet.md 6/4/2022

CHEATSHEET GIT

SWITCH

Rôle	Commande
Changer de branche	git switch <branche></branche>
Créer une branche et basculer sur celle-ci	git switch -c <branche></branche>

CHECKOUT

Rôle	Commande
Changer de branche	git checkout <branche></branche>
Créer une branche et basculer sur celle-ci	git checkout -b <branche></branche>

LOG

Rôle	Commande
Afficher un log jusqu'à HEAD	git log
Afficher un log jusqu'à HEAD et au dessus	git logall
Afficher des logs d'une ligne seulement	git logoneline

COMMIT

Rôle	Commande	Remarques
Stage et commit des fichiers	git commit -am " <message>"</message>	Cela ne remplace pas un git add . suivi d'un commit car le flag -a va stage seulement les fichiers déjà suivis alors que git add . va stage les changements sur les fichiers suivis mais aussi les nouveaux fichiers qui n'étaient pas encore suivi
Remplacer son précédent commit	git commit amend	Cette commande va remplacer l'ancien commit par un nouveau qui contiendra les changements qu'on a stage en plus de ceux précédemment commit. Le hash sera différent. Si on veut changer le nom du commit, on peut rajouter -m <message>. Au contraire, si on veut juste ajouter des changements en gardant l'ancien message, on peut ajouterno-edit</message>

INFOS COMPLÉMENTAIRES

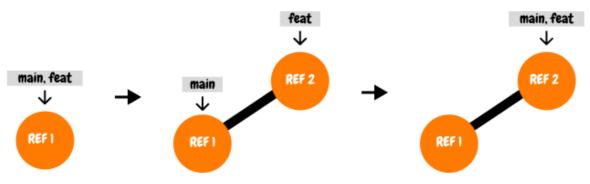
Merge

Quand on effectue un merge, il peut se passer trois choses :

git_cheatsheet.md 6/4/2022

Туре	Explications
merge fast forward	La branche sur laquelle on effectue le merge n'a pas eu de nouvelles modifications depuis que l'on travaille sur la branche et donc le merge sera un simple fast-forward , ce qui veut dire que le pointeur de la branche sur laquelle on merge va rattraper la branche visée
merge avec commit auto	Si la branche sur laquelle on effectue le merge a effectué un commit depuis





création d'une nouvelle branche feat, ayant la même référence que main

on effectue un commit sur feat, ce qui va créer une nouvelle référence on effectue maintenant un merge de la branche feat vers main. Étant donné que main n'a pas évolué entre temps, git va simplement changer sa référence et lui assigner celle que pointe feat