

# CHEATSHEET GIT

---

## SWITCH

Rôle	Commande
Changer de branche	<code>git switch &lt;branche&gt;</code>
Créer une branche et basculer sur celle-ci	<code>git switch -c &lt;branche&gt;</code>

## CHECKOUT

Rôle	Commande
Changer de branche	<code>git checkout &lt;branche&gt;</code>
Créer une branche et basculer sur celle-ci	<code>git checkout -b &lt;branche&gt;</code>

## LOG

Rôle	Commande
Afficher un log jusqu'à HEAD	<code>git log</code>
Afficher un log jusqu'à HEAD et au dessus	<code>git log --all</code>
Afficher des logs d'une ligne seulement	<code>git log --oneline</code>

## COMMIT

Rôle	Commande	Remarques
Stage et commit des fichiers	<code>git commit -am "&lt;message&gt;"</code>	Cela ne remplace pas un <code>git add</code> . suivi d'un commit car le flag <code>-a</code> va stage seulement les fichiers déjà suivis alors que <code>git add</code> . va stage les changements sur les fichiers suivis mais aussi les nouveaux fichiers qui n'étaient pas encore suivi
Remplacer son précédent commit	<code>git commit --amend</code>	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 <code>-m &lt;message&gt;</code> . Au contraire, si on veut juste ajouter des changements en gardant l'ancien message, on peut ajouter <code>--no-edit</code>

## INFOS COMPLÉMENTAIRES

---

### Merge

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

Type	Explications
<b>merge fast forward</b>	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 <b>fast-forward</b> , 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

### fast forward merge

