

### **Commandes utiles :**

git branch	-> voir les branches (courante et celles dans le cache)
git status	-> voir l'état des modifications sur la branche
git tag	-> voir l'ensemble des tag (local + remote)
git log	-> afficher les informations sur les commit de la branche
git log --oneline	
git log --stats	
git stash	-> stocker les modifications actuelles dans le « stash »
git stash list	
git stash pop	
git stash show@{<stash_index>}	
git stash apply	
git stash apply@{<stash_index>}	
git stash drop@{<stash_index>}	
git stash clear	
git checkout <branch_name>	-> aller sur <branch_name> + local change
git switch <branch_name>	-> aller sur <branch_name>
git switch -	-> revenir sur la branche précédente
git restore *file_name*	-> restaurer les fichiers sur la branche active
git restore --staged *file_name*	-> from HEAD

### **Créer une branche :**

git branch -m <branch\_name>  
git branch -M <branch\_name>  
git checkout -b <branch\_name>

### **Ajouter une branche sur la remote :**

git push -u origin <branch\_name>

### **Enregistrer des changements (sur une branche locale) :**

git add <file\_name>  
git add -p  
git add . -> **Attention**, ajoute TOUT

### **Ajouter des changement (sur une branche locale) :**

git commit  
git commit -m <commit\_message>  
git commit -am <commit\_message> -> ajoute ET enregistre

### **Stocker les changements sur la remote :**

git push  
git push -f -> à utiliser QUE dans certains cas

### Récupérer les changements de la remote :

git fetch	
git pull	
git pull --rebase	
git cherry-pick <commit_sha>	-> prendre un commit et l'ajouter sur la courante (penser à push)
git cherry-pick <c_sha1>.. <c_sha2&gt;< td=""><td>-&gt; c_sha1 est exclu</td></c_sha2&gt;<>	-> c_sha1 est exclu
git cherry-pick <c_sha1>^.. <c_sha2&gt;< td=""><td>-&gt; c_sha1 est inclu</td></c_sha2&gt;<>	-> c_sha1 est inclu
git cherry-pick --continue	
git cherry-pick --quit	
git cherry-pick --abort	
git rebase <branch_name>	-> rebase branch_name sur la courante (penser à push -f)
git rebase -i <branch_name>	
git merge <branch_name>	-> merger les changement de branch_name sur la courante (penser à push)
git merge --squash <branch_name>	-> regroupe les commits de branch_name en un seul (penser à push)
git merge --continue	
git merge --abort	

### Gestion des tags :

git tag -a <version>	-> création du tag
git tag -a <version> -m <message>	-> création du tag avec un message
git push <version> *origin*	-> stocker le tag sur la remote
git tag -d <version>	-> supprimer un tag local
git push --delete origin <version>	-> <b>supprimer un tag sur la remote</b>

### Suppression de branche :

git branch -d <branch_name>	-> supprimer une branche locale
git branch -D <branch_name>	-> supprimer une branche locale en cours de merge/rebase..
git push origin -d <branch_name>	-> <b>supprimer une branche sur la remote</b>

## Git flow :

