|  |  |
| --- | --- |
| **Git OBJECTIF** | **COMMANDE GIT** |
| créer un dépot local git | git init |
| valider une version de l'application | git commit -m "message" |
| Valider une version de l’application en remplaçant la dernière | git commit --amend -m "message" |
| se déplacer dans l'historique | git checkoutidentifiantDuCommit ou nomDuTag |
| créer une nouvelle branche à partir d'une version de l'historique | git checkoutidentifiantDuCommit -b nomDeLaNouvelleBranche |
| Afficher la liste des branches | git branch -a |
| Afficher uniquement la liste des branches distantes | git branch -r |
| afficher l'historique courant (la branche courante) | git log |
| Pour afficher tous les historiques (toutes les branches) | git log --branches |
| créer une branche après un déplacement dans l'historique | gitcheckout -b nomDeLaNouvelleBranche |
| Créer une branche nommée depuis un emplacement spécifié. Celle-ci ne contient pas d’historique | git checkout--orphan=nomDeLaNouvelleBrancheidCommit-ou-tag-ou-branche |
| consulter l'état du répertoire de travail (working directory) | git status |
| Parcourir et modifier l'historique depuis une version de l'historique | git rebase --interactive identifiantDuCommit ou nomDuTag |
| Parcourir et modifier l'historique depuis sa création | git rebase -i --root |
| créer un TAG associé à la version courante de l'application  | git tag nomDuTag |
| Créer un TAG associé à une version de l’historique | git tag nomDuTagidentifiantDuCommit |
| supprimer un TAG associé à une version de l'application | git tag nomDuTag -d |
| Envoyer l’ensemble des TAG sur un dépôt distant | git push origin --tags |
| envoyer son historique sur un dépôt distant | git push --repo=url |
| récupérer tout ou partie d'un dépôt distant (synchronisé) et réaliser une fusion avec l’existant | git pull |
| récupérer tout ou partie d'un dépôt distant (synchronisé) sans fusion | git fetch |
| récupérer le dernier commit d’undépôt distant (non synchronisé), d’historique différent et réaliser une fusion avec l’existantsans conserver l’historique du dépôt distant. | git pull --squash -–allow-unrelated-histories –-depth=1  |
| Cloner un dépôt local | git clone *<cheminDuDepotClone>* --no-hardlinks*<cheminDeDestinationDuClone>* |
| Lister les fichiers en conflits lors d’un merge | git diff –name-only –-diff-filter=U/\* alternatives \*/git status |
| Annuler une fusion en cours | git rebase --abort |
| Annuler un reset en cours | git reset 'HEAD@{1}' |
| Annuler une commande exécutée (et retrouver l’état correspondant)La valeur 1 correspond au nombre de commandes annulées | git reset --hard HEAD^1 |
| Afficher les informations relatives au dépôt distant configuré (URL, Branches et références locales configurées) | git remote show origin |
| Afficher la liste des actions précédemment exécutées.Permet d’annuler des commandes en s’appuyant sur la valeur de HEAD | git reflog |
| Afficher le contenu d’un d’un point de l’historique : répertoires et fichiers | git ls-treebranch/idCommit/tag |
| Afficher l’historique courant en utilisant une option de mise en forme | git log --pretty=format:"Commit id :%h %Cgreen %D %Creset %an " |
| Permet d’extraire un commit et de le fusionner dans l’historique courant | git cherry-pick id-commit |
|  | Sudo chown –R dev :www.data html |