git - commandes usuelles

CREATION

Clone le dépôt d'un projet sur le poste local
```
$ git clone <url_depot>
```

Crée le dépôt d'un nouveau projet sur le poste local
```
$ git init
```

MODIFICATIONS

Affiche les fichiers modifiés et l'index
```
$ git status
```

Affiche les modifications des fichiers
```
$ git diff
```

Ajoute tous les fichiers indiqués (<f1> <f2> ou <motif> ou <repertoire>) à l'index
```
$ git add <fichiers-indiqués>
exemple : $ git add src/*.c
exemple : $ git add .
exemple : $ git add f1.c f2.c
```

Choisit les modifications de <f1> ajoutées à l'index
```
$ git add -p <f1>
```

Crée un cliché des modifications enregistrées dans l'index
```
$ git commit -m <message>
```

Crée un cliché des modifications de tous les fichiers suivis
```
$ git commit -a -m <message>
```

Modifie le dernier cliché
⚠ Ne pas utiliser si le cliché a été publié sur le dépôt distant !
```
$ git commit --amend
```

ETIQUETTES

Etiquette le cliché actuel avec <nom>
```
$ git tag <nom>
```

Affiche les étiquettes satisfaisants une expression régulière <motif>
```
$ git tag -l <motif>
```

HISTORIQUE

Affiche les clichés à partir du plus récent
```
$ git log
```

Affiche les modifications de <fichier>
```
$ git log -p <fichier>
```

Affiche les intervenants sur <fichier>
```
$ git blame <fichier>
```

ANNULATIONS

Supprime toutes les modifications du répertoire de travail
```
$ git reset --hard HEAD
```

Supprime les modifications de <fichier>
```
$ git checkout HEAD <fichier>
```

Crée un cliché annulant les modifications de <cliche>
```
$ git revert <cliche>
```

Remet la branche actuelle au cliché <cliche> en supprimant les modifications ultérieures.
```
$ git reset --hard <cliche>
```

Remet la branche actuelle dans l'état de <cliche> en conservant les modifications ultérieures, sans l'index.
```
$ git reset <cliche>
```

Remet la branche actuelle dans l'état de <cliche> en conservant les modifications ultérieures et l'index.
```
$ git reset --keep <cliche>
```

DEPOTS DISTANTS

Affiche les dépôts distants
```
$ git remote -v
```

Affiche les informations d'un dépôt distant
```
$ git remote show <remote>
```

Ajoute le dépôt distant <nom> dont l'adresse est <url>
```
$ git remote add <nom> <url>
```

MISES A JOUR LOCALES

Télécharge sur le poste local les modifications de <distant>, sans les appliquer
```
$ git fetch <distant>
```

Télécharge sur le poste local les modifications de <distant> et les applique à <branche>
```
$ git pull <distant> <branche>
```

PUBLICATIONS

Publie les clichés locaux de <branche> sur un dépôt distant
```
$ git push <distant> <branche>
```

Publie les étiquettes sur le dépôt distant
```
$ git push <distant> --tags
```

BRANCHES

Affiche toutes les branches
```
$ git branch -av
```

Prend <branche> comme branche active
```
$ git switch <branche>
```

Crée une branche <nouvelle>
```
$ git branch <nouvelle>
```

Supprime <branche>
```
$ git branch -d <branche>
```

FUSIONS

Fusionne <branche> dans la branche active
```
$ git merge <branche>
```

Lance l'éditeur pour résoudre manuellement les conflits de fusion
```
$ git mergetool
```

Ajoute les résolutions de conflits à l'index et valide
```
$ git add <f1-ok> <f2-ok>
$ git rm <f3-pas-ok>
$ git commit -m <message>
```

Crée obligatoirement un cliché lors de la fusion
```
$ git merge <branche> -no-ff
```

REBASE

Rebase <branche> au cliché courant
⚠ Ne pas utiliser si le cliché a été publié sur le dépôt distant !
```
$ git rebase <branche>
```

Abandonne un rebase
```
$ git rebase --abort
```

Termine un rebase après la résolution des conflits
```
$ git rebase --continue
```

REFERENCES

Affiche la documentation d'une commande
```
$ git <commande> --help
```

Documentation officielle
```
https://git-scm.com/docs
```

Mémento interactif
```
https://ndpsoftware.com/git-cheatsheet.html
```