

### INFORMATIONS À DISTANCE

#### Cloner un référentiel

```
git clone <repository_url>
```

#### Obtenez des références et des objets distants

```
git fetch
```

#### Obtenir une branche distante (récupérer précédemment)

```
git checkout <remote_branch_name>
```

#### Récupérer modification distante d'un repo

```
git pull
```

#### Récupérer modification distante d'un repo et ajouter nos modifs

```
git pull --rebase
```

#### Modifications locales identiques et cachées

```
git pull -r --autostash
```

#### Envoyer les modifications au repository distant

```
git push origin <local_branch>
```

#### Envoyer les modifications au repository local vers distant

```
git push origin <local_branch>: <remote_branch>
```

#### Forcer les changements de poussée

```
git push -f origin <local_branch>
```

### DIRECTION

#### Créer une branche

```
git branch <branch_name>
```

#### Supprimer la branche locale

```
git branch -d <branch_name>
```

#### Suivre la branche distante

```
git branch --set-upstream-to origin / <remote_branch_name> <local_branch_name>
```

#### Déplacer de la branche

```
git checkout <branch_name>
```

#### Créer une branche et passer sur celle-ci

```
git checkout -b <branch_name>
```

#### Supprimer la branche distante

```
git push origin --delete <remote_branch_name>
```

### SAUVEGARDER LES MODIFICATIONS

### SAUVEGARDER LES MODIFICATIONS (cont)

#### Ajouter des fichiers au message de maintien de validation existant

```
git commit --amend --no-edit
```

#### Pas de modifications validées enregistrées rapidement dans la pile

```
git stash -u
```

#### Afficher toutes les cachettes

```
git stash list
```

#### Appliquer les modifications de la cachette

```
git stash apply
```

#### Appliquer la dernière cachette et retirer de la pile

```
git stash pop
```

### ANNULER LES CHANGEMENTS

#### Annuler les modifications de fichier hors du stage

```
git checkout <nom de fichier>
```

#### Sortez le fichier de la scène

```
git reset <nom de fichier>
```

#### Passer sur une branche et annuler toutes les modifications

```
git checkout -f <branch>
```

#### Annuler N derniers validations

```
git reset --hard HEAD ~ N
```

#### Annuler N derniers commits de maintien des changements dans stage

```
git reset HEAD ~ N
```

#### Annuler N derniers engagements en maintenant les modifs hors du stage

```
git reset --soft HEAD ~ N
```

#### Annuler les modifications N dernières validations (à distance)

```
git revert HEAD ~ N
```

### STATUT

### STATUT (cont)

```
statut git
```

#### Afficher les modifications de fichier

```
git diff <nom de fichier>
```

#### Afficher l'historique des validations de la branche actuelle

```
git log
```

### FUSION

#### Ajouter un commit d'une autre branche

```
git cherry-pick <commit_hash>
```

#### Fusionner les branches récursivement en créant un nouveau commit

```
git merge <branch_name>
```

#### Annuler la fusion

```
git merge --abort
```

#### Fusionner en plaçant le nôtre sur la branche de base

```
git rebase <base_branch_name>
```

#### Annuler le rebase

```
git rebase --abort
```

#### Changements de courbes

```
git rebase -i
```

**Ajouter des modifications de fichier à l'étape de validation**

*git add <nomfichier>*

**Ajouter toutes les modifications à la scène**

*git add -A*

**Créer un commit avec un message**

*git commit -m "<message>"*

**Afficher la branche actuelle**

*git branch*

**Afficher les succursales suivies locales et distantes**

*git branch -a*

**Liste des informations de branche en mode détaillé**

*git branch -vv*

**Afficher le statut de la succursale**



By **[deleted]**

[cheatography.com/deleted-108559/](https://cheatography.com/deleted-108559/)

Published 31st December, 2019.

Last updated 31st December, 2019.

Page 1 of 2.

---

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>