

### Configuration Initiale

*Définit votre nom :*

```
git config --global user.name " Nom "
```

*Définit votre email :*

```
git config --global user.email " ema il@ exe mpl e. c om"
```

*Affiche la configuration actuelle :*

```
git config --list
```

### Dépôts Locaux

*Crée un nouveau dépôt Git :*

```
git init
```

*Ajoute un fichier à la staging area :*

```
git add [fichier]
```

*Ajoute tous les fichiers modifiés :*

```
git commit -m " Mes sag e"
```

*Affiche l'état des fichiers (modifiés, indexés, etc.) :*

```
git status
```

*Affiche l'historique des commits :*

```
git log
```

*Historique simplifié :*

```
git log --oneline
```

### Branches

*Liste toutes les branches :*

```
git branch
```

*Crée une nouvelle branche :*

```
git branch [nom]
```

*Change de branche :*

```
git checkout [nom]
```

*Crée et change de branche :*

```
git checkout -b [nom]
```

*Fusionne une branche avec la branche actuelle :*

```
git merge [nom]
```

*Supprime une branche (-d pour forcer) :*

```
git branch -d [nom]
```

### Gestion des Conflits

### Annuler des Actions

*Annule des commits :*

```
git reset [commit] (options : --soft, --mixed, --hard)
```

*Annule les modifications d'un fichier non indexé :*

```
git checkout -- [fichier]
```

*Rétablit un fichier :*

```
git restore [fichier]
```

*Crée un nouveau commit annulant un commit précédent :*

```
git revert [commit]
```

### Dépôts Distants (GitHub Repositories)

*Clone un dépôt distant en local :*

```
git clone [url]
```

*Ajoute un dépôt distant :*

```
git remote add [nom] [url]
```

*Envoie les commits vers le dépôt distant :*

```
git push [remote] [branche]
```

*Récupère les mises à jour du dépôt distant :*

```
git pull
```

*Récupère les données distantes sans fusionner :*

```
git fetch
```

### Stash (Sauvegarde Temporaire)

*Sauvegarde les modifications non committées :*

```
git stash
```

*Récupère la dernière sauvegarde :*

```
git stash apply
```

*Liste toutes les sauvegardes :*

```
git stash list
```

*Supprime une sauvegarde :*

```
git stash drop
```

### Bonnes Pratiques

- **.gitignore** : Liste les fichiers/dossiers à ignorer (ex: node\_modules/, .env).
- **Messages de commit** : Soyez clairs et descriptifs.
- **Push régulier** : Synchronisez souvent avec le dépôt distant.

---

*Affiche les différences entre versions :*

git diff

*Ouvre un outil de résolution de conflits :*

git mergetool

**Après résolution de conflit , faire :**

git add [fichier] + git commit

---



By **HaffouzMohamedAmin**

[cheatography.com/haffouz mohamedamin/](https://cheatography.com/haffouz mohamedamin/)

Published 2nd February, 2025.

Last updated 21st September, 2026.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>