

# Git

Devenez responsable et adoptez les bons réflexes de versionning !

- [\[Git\] Cheat-sheet](#)

# [Git] Cheat-sheet

## Introduction

Git permet de faire du **versioning** notamment pour conserver et pouvoir restaurer son code dans le temps.



# git

## Configuration d'un dépôt

### Initialisation d'un dépôt

```
git init
```

### Configurer les identifiants

- Nom d'utilisateur :

```
git config --global user.name "<Username>"
```

- Email :

```
git config --global user.email "<MAIL@EMAIL.COM>"
```

## Configurer un dépôt distant

```
git remote add origin <REPOS_URL>
```

# Gestion d'un dépôt

## Ajouter les fichiers modifiés

```
git add .
```

## Faire un commit

```
git commit -m "<COMMIT_MESSAGE>"
```

## Faire un push

```
git push origin <BRANCH>
```

La branche principale se nomme **main**.

## Récupérer les fichiers d'un dépôt distant sans fusion

```
git fetch
```

## Récupérer les fichiers du dépôt distant

```
git pull origin <BRANCH>
```

La commande **git pull** est une combinaison de **git fetch** et **git merge**.

# Gestion des branches

## Créer une branche

```
git branch <BRANCH_NAME>
```

## Basculer sur une branche

```
git checkout <BRANCH_NAME>
```

## Lister les branches

```
git branch
```

## Fusionner une branche avec la branche actuelle

```
git merge <BRANCH_NAME>
```

# Gestion des commits

## Afficher l'historique des commits

```
git log
```

## Restaurer à l'état d'un commit

```
git reset <--hard|--soft> <COMMIT_ID>
```

L'argument **--soft** permet de restaurer seulement en local et vous devrez push pour appliquer les modifications contrairement à l'argument **--hard**.

# Autres commandes

## Cloner un dépôt

```
git clone <REPOS_URL>
```