

# Formation Git

---

Formateur : Nicolas Talle

- <https://github.com/Nicolab>
- <https://www.linkedin.com/in/ntalle/>

## Mise en pratique

### Installation

- Installer Git
- Installer extensions Gitlens dans VS code.

### git config

Configurer Git globalement:

```
git config --global user.name "Moi"
git config --global user.email "mon-email@example.org"
git config --global core.editor "nano" # ou vim
git config --global merge.tool vscode

git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

### hello-git

Créer un dépôt dans le dossier **hello-git**:

```
git init
```

[doc: git init](#)

Voir l'état actuel du dépôt:

```
git status
```

[doc: git status](#)

Ajouter les fichiers à l'index:

```
git add .
```

[doc: git add](#)

Enregistrer les modifications:

```
git commit -m "Initial commit"
```

Conseil: message court, explicite, à l'impératif.

[doc: git commit](#)

Voir l'historique:

```
git log
```

[doc: git log](#)

---

Pour ignorer des fichiers et dossiers en créant un fichier `.gitignore`.

[doc: .gitignore](#)

## Créer un compte Github

- Ajouter une clé SSH
- Pousser notre dépôt `hello-git`

## Tuto

Suivre le tuto :

- <https://www.pierre-giraud.com/git-github-apprendre-cours/>

## Collaboration

Créez des groupes de 2 ou 3. Forkez vos repos respectif, clonez en local, puis créez une branche portant votre prénom.

A l'intérieur de cette branche, modifiez le repo du collègue et envoyer une pull request.

Indiquez quelques précisions sur vos modifications dans la discussion de la pull request.

Le mainteneur doit merger les pull requests.

## Commandes utiles

- `git init` - Initialise ou reset un dépôt Git.
- `git status` - Voir l'état actuel.
- `git log` - Afficher les commits (historique).
- `git diff` - Afficher les différences depuis le derniers commit.
- `git add .` - Indexe (pour commiter) toutes les modifications du dossier courant.
- `git commit -m "Ma super feature"` - Enregistre les modifications dans l'historique.
- `git reset --soft HEAD~1` - Revenir en arrière d'un commit.
- `git merge develop` - Fusionne la branche *develop* dans la branche courante.
- `git branch -a` - Lister les branches.
- `git checkout` - Changer de branche.
- `git checkout -b` - Créer une branche.
- `git remote -v` - Lister les dépôts distant.
- `git fetch` - Récupérer toutes les branches.
- `git pull origin master` - Récupérer les dernières modifications de la branche master distante (origin).
- `git push origin master` - Pousser les dernières modifications sur la branche master distante (origin).