

# Site et installation

<https://git-scm.com/>

<https://git-scm.com/download/win>

## Download for Windows

[Click here to download](#) the latest **(2.46.0) 32-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **about 2 months ago**, on 2024-07-29.

Après installation, utiliser les terminal **git bash** ou tout autre terminal.

```
$ git config --global user.name "Charles Carnus"
```

```
$ git config --global user.email lycee@carnus.fr
```

```
$ git config --global init.default branch main
```

```
$ git config -h (pour le help)
```

```
$ git help config (pour le help dans le navigateur)
```

```
$ cd repertoire_de_travail (Exp_Git)
```

```
$ git init (initialiser le repertoire dans git) (un dossier caché .git a été ajouté au repertoire)
```

```
$ git status (état du repertoire) (Les fichiers ne sont pas ajoutés pour le moment)
```

```
$ git add index.html (pour ajouter le fichier index.html)
```

```
$ git status
```

```
$ git rm --cached index.html (pour le supprimer)
```

```
$ git status
```

Aller dans le dossier, créer un fichier nommé **.gitignore**

Puis insérer les commandes suivantes dans le fichier créé (pour ignorer tous les fichiers .txt) :

```
# ignore all .txt files
```

```
*.txt
```

```
$ git status (Le fichier ne doit pas apparaître)
```

```
$ git add -all (-A ou .)
```

```
$ git status
```

```
$ git commit -m "premier commit - commit de tous les fichiers dans le repository »
```

```
$ git status
```

Puis, apporter une modification au fichier index.html

```
$ git status (il doit signaler que le fichier index.html a été modifié)
```

```
$ git diff (pour voir la différence entre les deux versions du fichier)
```

```
$ git add index.html
```

```
$ git restore --staged index.html
```

```
$ git commit -a -m "mise à jour du texte"
```

```
$ git rm style.css
```

```
$ git status
```

```
$ git restore style.css
```

```
$ git mv "logo.png" "logoCIEL.png" (pour modifier le nom d'un fichier)
```

```
$ git commit -m "Modification du nom du fichier html" (Réellement, on a modifié le nom de l'image)
```

```
$ git log
```

```
$ git log --oneline
```

```
$ git commit -m "Modification du nom de l'image" --amend
```

```
$ git log --oneline
```

```
$ git log -p
```

```
$ git help log
```

```
$ git log --oneline
```

```
$ git rebase -i --root (ctrl Q pour quitter)
```

```
% $ git reset LeNuméroDuChangementDeTexteDansHTML
```

Créer une nouvelle branche : (pour changer le logo à utiliser dans le fichier html)

```
$ git branch ChangeLogo
```

```
$ git branch (pour voir toutes les branches et la branche actuelle (main))
```

```
$ git switch ChangeLogo (pour aller dans la nouvelle branche)
```

```
$ git branch
```

Modifier le fichier html en changeant le logo

```
$ git status
```

```
$ git commit -a -m "mise à jour du logo"
```

```
$ git switch main (pas de modifications)
```

```
$ git merge -m "Merge changement de logo dans main" ChangeLogo
```

```
$ git branch -d ChangeLogo (pour supprimer la branche)
```

```
$ git branch
```

```
$ git switch -c ChangeTexte (pour créer une nouvelle branche et switcher dans celle-ci)
```

```
$ git branch
```

Modifier du texte dans le fichier html

```
$ git commit -a -m "Modification html"
```

```
$ git switch main
```

Modifier le fichier html

```
$ git commit -a -m "Modification html"
```

```
$ git merge ChangeTexte (un conflit)
```

Supprimer les partie inutiles dans le fichier html

```
$ git commit -a -m "Modification html"
```

