

4. Modifier un dépôt git

Ajouter ou modifier des fichiers

On a vu [Dans le chapitre précédent](#) comment ajouter et modifier des fichiers, avec `git add`, et `git commit`. On peut également condenser les 2 actions en utilisant `git commit -a`.

Supprimer un fichier d'un projet

Pour supprimer un fichier d'un dossier on peut utiliser la commande `rm` qui prend en argument le nom du fichier à supprimer.

```
rm [nom_du_fichier]
```

Supprimer un fichier initialement suivi par git ne suffira pas pour qu'il l'oublie.

```
touch test2.txt
git add test2.txt
rm test2.txt
git status
On branch master
```

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
new file: test-2.txt

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: test-2.txt

Pour forcer git à oublier le fichier on doit utiliser la commande `git rm`.

```
git rm test2.txt
git status
On branch master
nothing to commit, working tree clean
```

Pour simplement exclure un fichier du suivi git tout en le gardant dans le projet, on va utiliser la commande `git rm` avec l'option `-cached`.

```
touch test2.txt
git add test2.txt
git commit -m "Ajout du fichier 2"
[master 8cc6122] ajout du fichier 2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test2.txt
git rm --cached test2.txt
git status
```

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
deleted: test2.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)
test2.txt

Ici, le fichier `test2.txt` a bien été exclu de git mais existe toujours dans le projet.

Supprimer un fichier de la staging area

[Pour rappel](#), la *staging area* va regrouper tous les fichiers qui seront enregistrés dans le prochain `commit`. Pour retirer un fichier de cette zone, on peut utiliser la commande `git reset`.

À la différence de `git rm`, le fichier va garder son statut suivi.

```
touch test3.txt
```

```
git add test3.txt
```

```
git status
```

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
new file: test3.txt

```
git reset test3.txt
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be
  committed)
    test3.txt
```

Empêcher l'indexation de certains fichiers

Certains fichiers de projet sont des fichiers de configuration, ou des fichiers sensibles. On peut bloquer l'indexation de certains fichiers en créant dans notre dossier un fichier `.gitignore`, et en y mettant tous les fichiers qu'on désire ignorer. Ce fichier accepte également des `fileglob`.

Exercice

Dans le dossier de l'exercice précédent, supprimer "fichier1.txt"

Effectuer un `git status`

Supprimer effectivement "fichier1.txt" de la *staging area*

Recréer "fichier1.txt" et un fichier ".gitignore"

Faire en sorte que "fichier1.txt" soit ignoré par git