

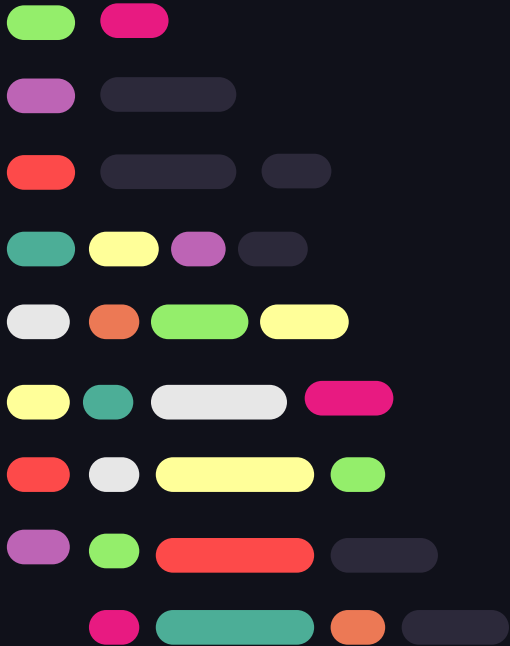
**OOPSIE!**



Revenir en arrière quand on a tout KC



# Retirer de la staging area



```
git add <path/to/file>
```

*# J'ai add un fichier qu'il fallait pas ! 😱*

*# Retirer de la staging area*

```
git reset <path/to/file>
```

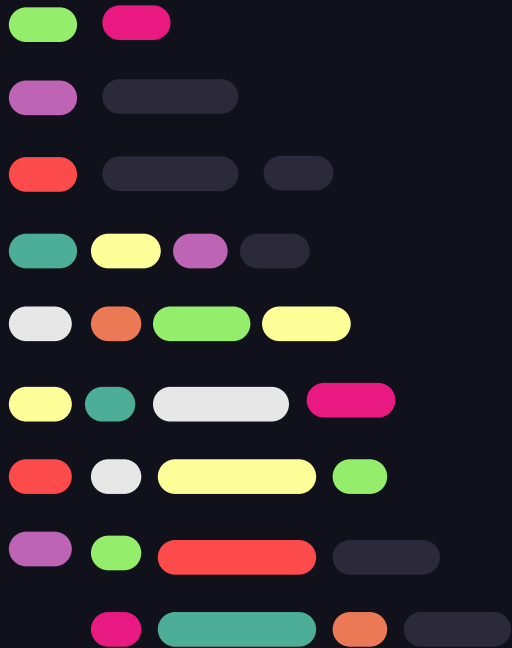
*# ou (c'est équivalent)*

```
git restore --staged <path/to/file>
```

*# On vérifie que tout est ok*

```
git status
```

# Retirer d'un commit local

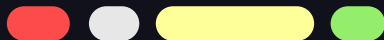


```
git add <path/to/file>
git commit -m "Ajout de file par erreur"
# J'ai commit un fichier qu'il fallait pas ! 🤖

# Retirer le fichier du dernier commit réalisé
git reset <path/to/file>

# Modifier le dernier commit
git commit --amend
```

# Réécrire un commit déjà pushé



```
git add <path/to/file>
```

```
git commit -m "Ajout de file par erreur"
```

```
git push
```

*# J'ai pushé un fichier qu'il fallait pas ! 🤪*

*# Retirer le fichier du dernier commit réalisé*

```
git reset <path/to/file>
```

*# Réécrire le dernier commit pushé*

```
git commit --amend
```

```
git push --force
```

*# ⚠️ **--force** peut créer des conflits si d'autres ont déjà récupéré la version précédente du commit*

# Ajouter à un commit local



*# J'ai oublié un fichier à mon commit ! 🤪*

*# Indexation du fichier*

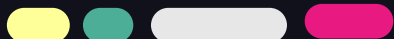
```
git add <path/to/file>
```

*# Modifier le dernier commit*

```
git commit --amend
```



# Désindexer des fichiers

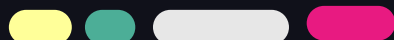


```
git add .  
git commit -m "Indexation de .env/ par erreur"  
git push  
# J'ai oublié de l'ajouter au .gitignore ! 🤪
```

```
# Désindexation du dossier  
git rm --cached .env/
```

```
# Créer un commit de correction  
git commit -m "Désindexation de .env/"  
git push
```

# Modifier le message d'un commit



```
git commit -m "Message de commit nul"
```

```
# Mon message de commit est nul ! 😱
```

```
# Modifier le message du dernier commit réalisé
```

```
git commit --amend -m "Message de commit mieux"
```

# Réécrire un message déjà pushé



```
git commit -m "Message de commit nul"  
# Mon message de commit est nul ! 🤪
```

```
# Modifier le message du dernier commit réalisé  
git commit --amend -m "Message de commit mieux"
```

```
# Forcer la réécriture du commit sur GitHub  
git push --force
```

```
# ⚠️ --force peut créer des conflits si d'autres  
ont déjà récupéré la version précédente du commit
```





# Annuler les modifications



*# Effacer les mods effectuées sur un fichier  
# depuis le dernier commit*

```
git restore <path/to/file>
```

*# Sur tous les fichiers*

```
git restore .
```

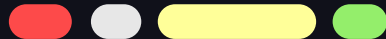
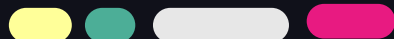
*# Mettre de côté des modifications non validées*

```
git stash push <path/to/file>
```

*# Récupérer des modifications mises de côté*

```
git stash pop
```

# Supprimer un commit



*# Afficher l'historique*

```
git log --oneline
```

abc1234 Dernier commit (à supprimer)

def5678 Commit précédent

ghi9101 Commit plus ancien

*# Retirer tous les commits depuis def5678*

*# et **conserver** les modifications en local*

```
git reset --soft def5678
```

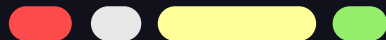
*# et **supprimer** les modifications*

```
git reset --hard def5678
```

*# Forcer la suppression des commit sur GitHub*

```
git push --force
```

# Voir comment c'était avant



*# Afficher l'historique*

```
git log --oneline
```

abc1234 Dernier commit

def5678 Commit précédent

ghi9101 Commit plus ancien

*# Se déplacer au commit def5678*

```
git checkout def5678
```

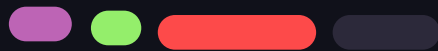
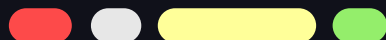
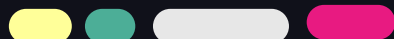
*# Créer une nouvelle branche à partir de là*

```
git switch -b <branch-name>
```

*# Ou revenir à sa branche d'origine*

```
git switch <original-branch-name>
```

# Supprimer un merge



*# Afficher l'historique*

```
git log --oneline
```

```
abc1234 Dernier commit
```

```
def5678 Commit de MERGE
```

*# Reset le merge*

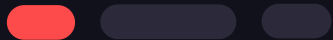
```
git revert -n -m 1 def5678
```

*# Reset le merge*

```
git commit -m "Revert changes from merge def5678"
```



# Supprimer une branche

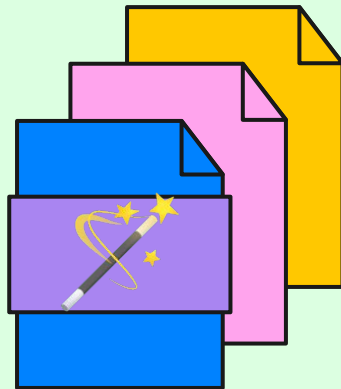
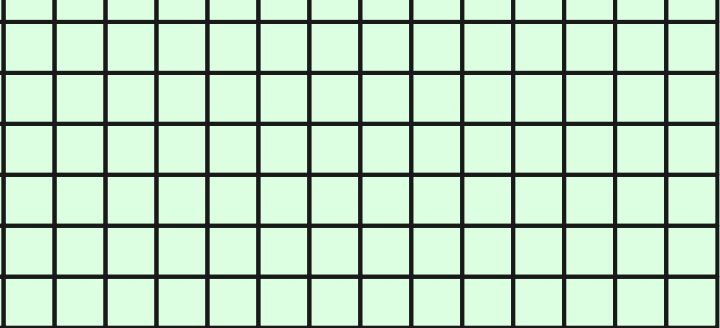


*# En local*

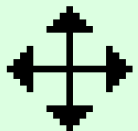
```
git branch -d <branch-name>
```

*# Sur le remote*

```
git push origin --delete <branch-name>
```



# TIPS & TRICKS



Commandes & outils magiques



# Fusionner un seul commit



*# Sur la branche où se trouve le commit*

*# Afficher l'historique*

```
git log --oneline
```

abc1234 Dernier commit

def5678 Commit à intégrer

*# Se déplacer sur la branche où intégrer le commit*

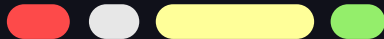
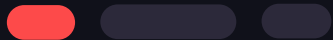
```
git switch <branch-name>
```

*# Intégrer seulement le commit*

```
git cherry-pick def5678
```



# Qu'est-ce qui est merged ?



*# Lister les branches mergées dans main*  
`git branch --merged main`

*# Lister les branches mergées dans HEAD*  
*# (dernier commit de la branche courante)*  
`git branch --merged`