

# Exercice sur les bases de git

Se familiariser avec le vocabulaire et les commande avant de commencer l'exercice

**objectif:** Découvrir git, et ses commandes de base:

- Voir la difference des fichiers entre deux commit
- Gérer le commits:
  - Modifier le dernier message du commit
  - Annuler un commit
  - Annuler les modification avant un commit
  - Annuler un fichier avant commit

## Exercice

1. Aller dans `site/lab` et créer un dossier `testGit2`, aller dedans.
2. Initialiser le dossier avec git
3. Créer un fichier `index.html`, l'ajouter à git et le commiter.
4. Modifier le fichier `index.html` en ajoutant une phrase, l'ajouter à git et le commiter avec le message "blabla"
5. afficher les logs
6. Maintenant on se rend compte que le message du dernier commit n'est pas pertinent, on va donc le modifier.
  - Modifier le message du dernier commit par un message plus pertinent du genre "ajout d'une première phrase" (**voir git commit --amend**)
7. Finalement, on veut garder nos changements du fichier `index.html` mais on veut annuler le commit:
  - Annuler le dernier commit tout en gardant les modifications (**voir git reset**)
  - afficher le status et les logs
8. Ajouter une autre phrase à `index.html` et commiter avec le message "troisieme modif", afficher status et logs (on a que deux commit, celui "ajout d'une premiere phrase" à été annuler)
9. Maintenant, on va annuler le dernier commit **ainsi** que les modifications effectuée:
  - Annuler tous les changement du dernier commit (**voir git reset**)
  - afficher le status et les logs
  - ouvrir `index.html` -> retour à l'etat du premier commit!
10. Modifier `index.html` (qui doit être vide ) en ajoutant une phrase, puis commiter
11. effectuer cette opération deux fois supplémentaire
12. On a maintenant trois phrase dans `index.html` et 4 commit
13. Modifier `index.html` en ajoutant un paragraphe > attention: ne pas commiter
14. afficher le status et les logs
15. Afficher les changement du fichier `index.html` (**voir git diff**)
16. On va maintenant annuler les modifications avant le commit (**voir git checkout nomFichier**)
17. afficher le status (Que remarquez-vous ?)