1. Créer un dossier appelé learn_git.

```
● ● ●                    📁 learn_git — -zsh — 80×18
[7maksacodpc@7MAKSACODs-MacBook-Pro desktop % mkdir learn_git
```

2. Cd (changer de répertoire) dans le dossier learn_git.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro desktop % cd learn_git
```

3. Créer un fichier appelé third.txt.

```
7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % touch third.txt
```

4. Initialiser un dépôt git vide.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git init
Initialized empty Git repository in /Users/7maksacodpc/Desktop/learn_git/.git/
```

5. Ajouter third.txt à la staging area.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git add third.txt
```

6. Commiter avec le message "adding third.txt".

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git commit -m "adding third.txt"
[master (root-commit) 6fde1c5] adding third.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 third.txt
```

7. Vérifiez votre validation avec git log.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git log
commit 6fde1c5ec3e101ff5306146b60ed5dd17b97307b (HEAD -> master)
Author: Yves Placide <kouamekoffiyves57@gmail.com>
Date:   Sat Nov 30 17:03:36 2024 +0000

    adding third.txt
```

8. Créez un autre fichier appelé fourth.txt.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % touch fourth.txt
```

9. Ajoutez fourth.txt à la staging area.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git add fourth.txt
```

10. Effectuez un commit avec le message "adding fourth.txt".

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git commit -m "adding fourth.txt"

[master cac6e92] adding fourth.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fourth.txt
```

11. Enlevez le fichier third.txt.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % rm -rf third.txt
```
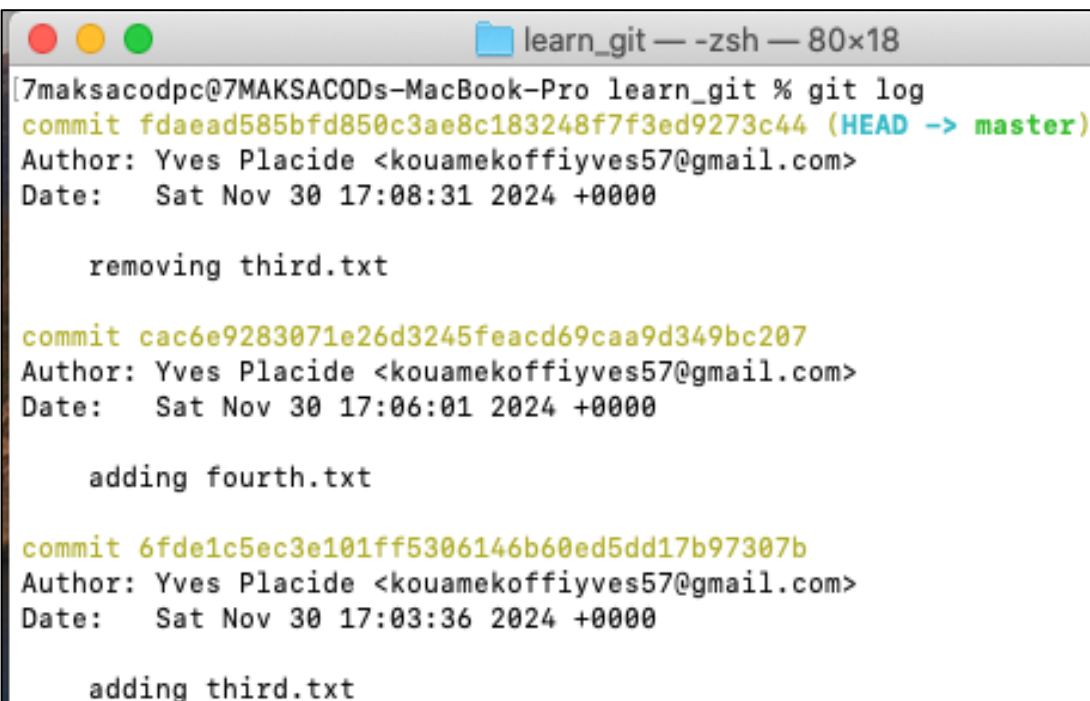
12. Ajoutez cette modification à la staging area. (En utilisant la commande "git add .")

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git add .
```

13. Commiter avec le message "removing third.txt".

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git commit -m "removing third.txt"
"
[master fdaead5] removing third.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 third.txt
```

14. Vérifiez vos commits en utilisant git log.

```
learn_git — -zsh — 80×18
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git log
commit fdaead585bfd850c3ae8c183248f7f3ed9273c44 (HEAD -> master)
Author: Yves Placide <kouamekoffiyves57@gmail.com>
Date:   Sat Nov 30 17:08:31 2024 +0000

    removing third.txt

commit cac6e9283071e26d3245feacd69caa9d349bc207
Author: Yves Placide <kouamekoffiyves57@gmail.com>
Date:   Sat Nov 30 17:06:01 2024 +0000

    adding fourth.txt

commit 6fde1c5ec3e101ff5306146b60ed5dd17b97307b
Author: Yves Placide <kouamekoffiyves57@gmail.com>
Date:   Sat Nov 30 17:03:36 2024 +0000

    adding third.txt
```
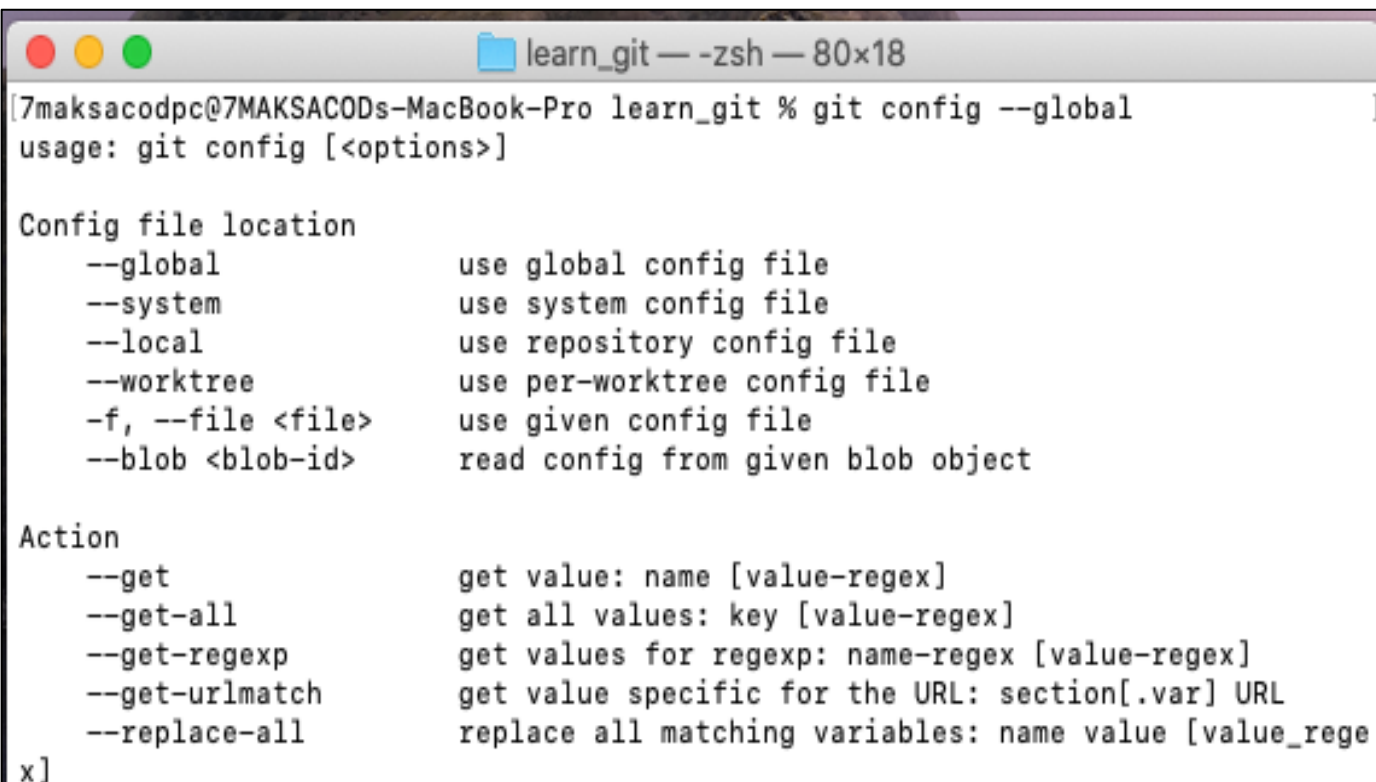
15. Changez vos paramètres globaux pour core.pager=cat - vous pouvez en savoir plus ici.

```
7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git config --global core.pager ca
t
```

16. Écrivez la commande appropriée pour lister toutes les configurations globales pour git sur votre machine.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git config --list
credential.helper=osxkeychain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Yves Placide
user.email=kouamekoffiyves57@gmail.com
core.pager=cat
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
7maksacodpc@7MAKSACODs-MacBook-Pro learn_git %
```

17. Vous pouvez taper git config --global pour savoir comment faire.

```
[7maksacodpc@7MAKSACODs-MacBook-Pro learn_git % git config --global
usage: git config [<options>]

Config file location
    --global              use global config file
    --system              use system config file
    --local               use repository config file
    --worktree            use per-worktree config file
    -f, --file <file>     use given config file
    --blob <blob-id>      read config from given blob object

Action
    --get                 get value: name [value-regex]
    --get-all             get all values: key [value-regex]
    --get-regexp          get values for regexp: name-regex [value-regex]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value_rege
x]
```