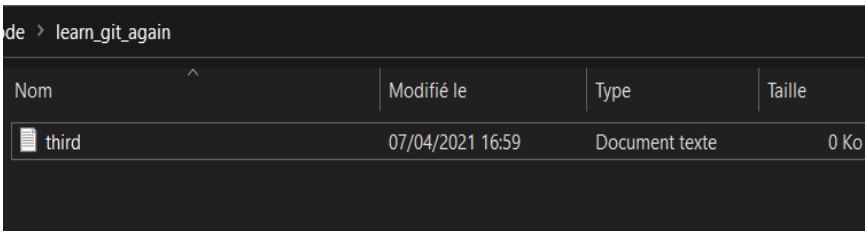
1. Create a folder called learn_git_again.
2. cd into the learn_git_again folder.
3. Create a file called third.txt.



4. Initialize an empty git repository.

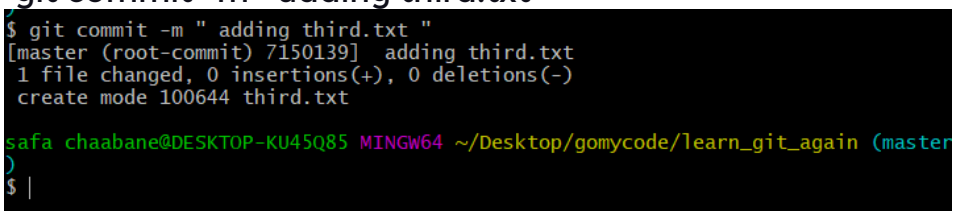
```
git init
```

5. Add third.txt to the staging area.

```
git add third.txt
```

6. Commit with the message "adding third.txt".

```
git commit -m " adding third.txt "
```



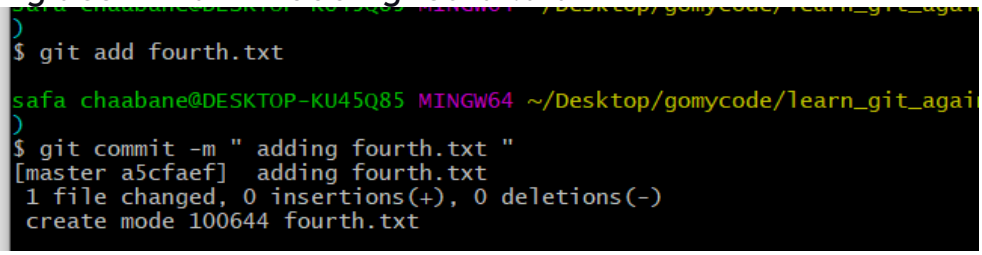
7. Check out your commit with git log.

8. Create another file called fourth.txt.
9. Add fourth.txt to the staging area.

git add fourth.txt

10. Commit with the message "adding fourth.txt"

git commit -m " adding fourth.txt "



11. Remove the third.txt file

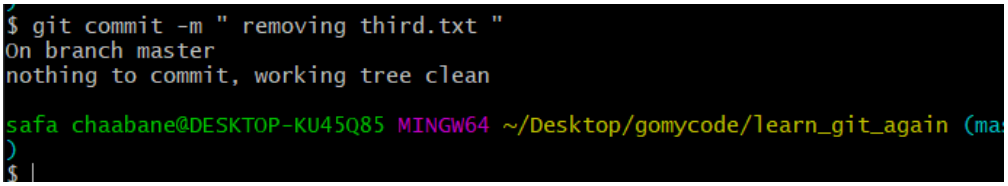
git rm --cached third.txt

12. Add this change to the staging area

git add .

13. Commit with the message "removing third.txt"

git commit -m " removing third.txt "



14. Check out your commits using git log

```
safa chaabane@DESKTOP-KU45Q85 MINGW64 ~/Desktop/gomycode/learn_git_again (mast
)
$ git log
commit a5cfaef6b0571c65308179b9c9e7704b5cf7eaae (HEAD -> master)
Author: safachaabane <safa1chaabane@gmail.com>
Date:   Wed Apr 7 22:46:22 2021 +0100

    adding fourth.txt

commit 71501397cd4cd91cf9b69263ea34db83c4397e01
Author: safachaabane <safa1chaabane@gmail.com>
Date:   Wed Apr 7 22:30:23 2021 +0100

    adding third.txt

safa chaabane@DESKTOP-KU45Q85 MINGW64 ~/Desktop/gomycode/learn_git_again (mast
)
```

15. Change your global setting to core.pager=cat

    git config --global --replace-all core.pager cat

```
)
$ git config --global --replace-all core.pager cat

safa chaabane@DESKTOP-KU45Q85 MINGW64 ~/Desktop/gomycode/learn_git_again (master
)
```

16. Write the command to list all the global configurations for git on your machine.

    git config --list --show-origin

```
)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-proces
s
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/ming
w64/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager-core
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttp
path=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/safa chaabane/.gitconfig  user.name=safachaabane
file:C:/Users/safa chaabane/.gitconfig  user.email=safa1chaabane@gmail.com
file:C:/Users/safa chaabane/.gitconfig  core.pager=cat
file:.git/config        core.repositoryformatversion=0
file:.git/config        core.filemode=false
file:.git/config        core.bare=false
file:.git/config        core.logallrefupdates=true
file:.git/config        core.symlinks=false
file:.git/config        core.ignorecase=true

safa chaabane@DESKTOP-KU45Q85 MINGW64 ~/Desktop/gomycode/learn_git_again (master
)
```

17. You can type git config --global to find out how to do this

```
$ 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-pattern]
    --get-all             get all values: key [value-pattern]
    --get-regexp          get values for regexp: name-regex [value-pattern]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value-pattern]
    --add                 add a new variable: name value
    --unset               remove a variable: name [value-pattern]
    --unset-all           remove all matches: name [value-pattern]
    --rename-section      rename section: old-name new-name
    --remove-section      remove a section: name
    -l, --list            list all
    --fixed-value         use string equality when comparing values to 'value-pattern'
    -e, --edit            open an editor
    --get-color           find the color configured: slot [default]
    --get-colorbool       find the color setting: slot [stdout-is-tty]

Type
    -t, --type <>         value is given this type
    --bool                value is "true" or "false"
    --int                 value is decimal number
    --bool-or-int         value is --bool or --int
    --bool-or-str         value is --bool or string
    --path                value is a path (file or directory name)
    --expiry-date         value is an expiry date

Other
    -z, --null            terminate values with NUL byte
    --name-only           show variable names only
    --includes            respect include directives on lookup
    --show-origin         show origin of config (file, standard input, blob, command line)
    --show-scope          show scope of config (worktree, local, global, system, command)
    --default <value>     with --get, use default value when missing entry
```