

Git checkpoint: learn_git_Insaf

Instructions

1. Create a folder called learn_git.

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git (main)
$ mkdir learn_git_Insaf
```

2. Cd (change directory) into the learn_git folder.
3. Create a file called third.txt.
4. Initialize an empty git repository.
5. Add third.txt to the staging area.

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git (main)
$ cd learn_git_Insaf

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (main)
$ touch third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (main)
$ git init
Initialized empty Git repository in C:/Users/EliteBook/D
esktop/Git/learn_git_Insaf/.git/

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git add third.txt
```

6. Commit with the message "adding third.txt".
7. Check out your commit with git log.

```

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git commit -m "adding third.txt"
[master (root-commit) 57138e5] adding third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git log --oneline
57138e5 (HEAD -> master) adding third.txt

```

8. Create another file called fourth.txt.
9. Add fourth.txt to the staging area.
10. Commit with the message "adding fourth.txt"
11. Remove the third.txt file.
12. Add this change to the staging area. (Using the command "git add . "

```

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ touch fourth.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git add fourth.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git commit -m "adding fourth.txt"
[master 71cadb2] adding fourth.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fourth.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ rm third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git add .

```

13. Commit with the message "removing third.txt".
14. Check out your commits using git log.
15. Change your global settings to core.pager=cat - you can read more about that here.

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

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git commit -m "removing third.txt"
[master c2ad629] removing third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git log --oneline
c2ad629 (HEAD -> master) removing third.txt
71cadb2 adding fourth.txt
57138e5 adding third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git config --global core.pager cat

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git config --list --global
user.name=Insaf
user.email=insaf.guedidi@gmail.com
core.pager=cat
```

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

```

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git config --global --help
usage: git config list [<file-option>] [<display-option>]
[  
  --includes]
  or: git config get [<file-option>] [<display-option>]
  [--includes] [--all] [--regexp] [--value=<value>] [--fi
xed-value] [--default=<default>] <name>
  or: git config set [<file-option>] [--type=<type>] [--
all] [--value=<value>] [--fixed-value] <name> <value>
  or: git config unset [<file-option>] [--all] [--value
=<value>] [--fixed-value] <name>
  or: git config rename-section [<file-option>] <old-na
me> <new-name>
  or: git config remove-section [<file-option>] <name>
  or: git config edit [<file-option>]
  or: git config [<file-option>] --get-colorbool <name>
  [<stdout-is-tty>]

Config file location
  --[no-]global          use global config file
  --[no-]system          use system config file
  --[no-]local           use repository config file
  --[no-]worktree        use per-worktree config file
  -f, --[no-]file <file> use given config file
  --[no-]blob <blob-id> read config from given blob ob
ject

Action
  --get                  get value: name [<value-patter
n>]
  --get-all             get all values: key [<value-pa
ttern>]
  --get-regexp           get values for regexp: name-re
gex [<value-pattern>]
  --get-urlmatch         get value specific for the URL

```

Etc.

After that I created a repository in GitHub with same project name then,

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git remote add origin https://github.com/insaf-guedidi
/learn_git_Insaf.git

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git remote -v
origin https://github.com/insaf-guedidi/learn_git_Insaf
.git (fetch)
origin https://github.com/insaf-guedidi/learn_git_Insaf
.git (push)

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git add .

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git commit -m "Initial commit with third.txt and fourt
h.txt"
On branch master
nothing to commit, working tree clean
```

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (master)
$ git branch -M main
git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 639 bytes | 106.00 KiB/s, d
one.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (fr
om 0)
To https://github.com/insaf-guedidi/learn_git_Insaf.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (main)
$ git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (main)
$ git log --oneline
c2ad629 (HEAD -> main, origin/main) removing third.txt
71cadb2 adding fourth.txt
57138e5 adding third.txt

EliteBook@InsafGuedidi MINGW64 ~/Desktop/Git/learn_git_I
nsaf (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Then I added the output of the checkpoint (a pdf file) in the folder, I will upload the output of this in a separate photo in GitHub