

Git & GitHub Checkpoint

Ranim ben chaouacha

What You're Aiming For

Now that you have learned the basics of Git workflow, try running through it a couple of times on your own (see instructions).

At this checkpoint, you are asked to take screenshots for each step in the instructions section, then push these screenshots to the GitHub repository.

Instructions

1. Create a folder called learn_git.
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.
6. Commit with the message "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.
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 .")
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.
17. You can type git config --global to find out how to do this.

```
C:\Users\PC>mkdir learn_git
```

```
C:\Users\PC>cd learn_git
```

```
C:\Users\PC\learn_git>cd> third.txt
```

```
C:\Users\PC\learn_git>git init
```

```
Initialized empty Git repository in C:/Users/PC/learn_git/.git/
```

```
C:\Users\PC\learn_git>git add third.txt
```

```
C:\Users\PC\learn_git>git commit -m "adding third.txt"
```

```
[master (root-commit) aab212f] adding third.txt
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 third.txt
```

```
C:\Users\PC\learn_git>git log
```

```
commit aab212fadd0fcc5dcb265256abd34d10ab702864 (HEAD -> master)
```

```
Author: Ranimbenchauacha <ranimbenchauacha093@gmail.com>
```

```
Date: Tue Jul 9 23:15:30 2024 +0100
```

```
    adding third.txt
```



```
C:\Users\PC\learn_git>git rm third.txt
rm 'third.txt'
```

```
C:\Users\PC\learn_git>git add .
```

```
C:\Users\PC\learn_git>git commit -m "removing third.txt"
[master 88985eb] removing third.txt
1 file changed, 1 deletion(-)
delete mode 100644 third.txt
```

```
C:\Users\PC\learn_git>git log
commit 88985eb155ba087cf701168e8b38af26b9f8c1fe (HEAD -> master)
Author: Ranimbenchouacha <ranimbenchouacha093@gmail.com>
Date: Tue Jul 9 23:21:43 2024 +0100
```

```
    removing third.txt
```

```
commit aab212fadd0fcc5dcb265256abd34d10ab702864
Author: Ranimbenchouacha <ranimbenchouacha093@gmail.com>
Date: Tue Jul 9 23:15:30 2024 +0100
```

```
    adding third.txt
```

```
C:\Users\PC\learn_git>git config --global core.pager "cat -"
```

```
C:\Users\PC\learn_git>git config --global --list
user.email=ranimbenchouacha093@gmail.com
user.name=Ranimbenchouacha
core.pager=cat -
```

```
C:\Users\PC\learn_git>git config --global
usage: git config [<options>]
```

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 object
```

Action

```
--[no-]get          get value: name [value-pattern]
--[no-]get-all      get all values: key [value-pattern]
--[no-]get-regexp    get values for regexp: name-regex [value-pattern]
--[no-]get-urlmatch  get value specific for the URL: section[.var] URL
--[no-]replace-all  replace all matching variables: name value [value-pattern]
--[no-]add           add a new variable: name value
--[no-]unset         remove a variable: name [value-pattern]
--[no-]unset-all    remove all matches: name [value-pattern]
--[no-]rename-section rename section: old-name new-name
--[no-]remove-section remove a section: name
-l, --[no-]list      list all
--[no-]fixed-value   use string equality when comparing values to 'value-pattern'
-e, --[no-]edit      open an editor
--[no-]get-color     find the color configured: slot [default]
--[no-]get-colorbool find the color setting: slot [stdout-is-tty]
```

Type

```
-t, --[no-]type <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, --[no-]null      terminate values with NUL byte
--[no-]name-only     show variable names only
--[no-]includes      respect include directives on lookup
--[no-]show-origin   show origin of config (file, standard input, blob, command line)
--[no-]show-scope    show scope of config (worktree, local, global, system, command)
--[no-]default <value>
                    with --get, use default value when missing entry
--[no-]comment <value>
                    human-readable comment string (# will be prepended as needed)
```