

# Learn Git - Basic commands

## 1-Créez un dossier appelé learn\_git:

Pour créer un dossier on utilise la commande < mkdir > ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~  
$ mkdir learn_git
```

## 2-Changer de répertoire:

On utilise la commande < cd > pour changer de répertoire ,et pour se placer dans le dossier qu'on a crée on fait < cd + nomDossier > = < cd Learn\_git > ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~  
$ cd learn_git  
  
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git  
$
```

## 3-Créer un fichier appelé third.txt :

La commande < touch > nous permet de créer le fichier ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git  
$ touch third.txt
```

## 4-Initialiser un dépôt git vide :

Il faut savoir qu'avant commencer d'utiliser git ,on doit d'abord initialiser notre dépôt et pour cela on tape la commande < git init > :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git  
$ git init  
Initialized empty Git repository in C:/Users/maria/learn_git/.git/  
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)  
$
```

## 5-Ajouter third.txt à la zone d'essai :

Pour l'ajouter ,c'est à l'aide de : < git add + nomFichier > = < git add third.txt > ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)  
$ git add third.txt
```

## 6-Effectuer un commit avec le message "adding third.txt" :

La commande < git commit -m " message" > pour effectuer un commit et sauvegarder notre travail ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git commit -m "adding third.txt"
[master (root-commit) 5effa5d] adding third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third.txt
```

## 7-Vérifier votre commit avec git log :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git log
commit 5effa5d4b7641f6fe69411afc70e1f1264d4307d (HEAD -> master)
Author: Maria Chiheb <airamchiheb@gmail.com>
Date:   Fri Jan 26 02:20:47 2024 +0100

    adding third.txt
```

## 8-Créer un autre fichier appelé fourth.txt :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ touch fourth.txt
```

## 9-Ajouter fourth.txt à la zone de préparation :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git add fourth.txt
```

## 10-Livrer avec le message "adding fourth.txt" :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git commit -m "adding fourth.txt"
[master eafe3e6] adding fourth.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fourth.txt
```

## 11-Supprimez le fichier third.txt :

Pour supprimer fichier on exécute la commande `< git rm + nomFichier >` ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git rm third.txt
rm 'third.txt'
```

## 12-Ajoutez cette modification à la zone de transit. (En utilisant la commande "git add.") :

`< git add . >` c'est une commande qui permet d'ajouter tous les fichiers du dépôt , pas non seulement un ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git add .
```

## 13-Commitez avec le message "removing third.txt" :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git commit -m "removing third.txt"
[master 4e13f36] removing third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 third.txt
```

## 14- Vérifiez vos commits à l'aide de git log :

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git log
commit 4e13f36883d97bef2f0879298fd3dd0cb44c3d40 (HEAD -> master)
Author: Maria Chiheb <airamchiheb@gmail.com>
Date:   Fri Jan 26 03:03:23 2024 +0100

    removing third.txt

commit 12f02426b1044a1844616c842e54ac3088d4c7a1
Author: Maria Chiheb <airamchiheb@gmail.com>
Date:   Fri Jan 26 02:57:54 2024 +0100

    adding fourth.txt

commit 999ac5f2c9f5a8068848c396e2728f3a4ff0aceb
Author: Maria Chiheb <airamchiheb@gmail.com>
Date:   Fri Jan 26 02:56:44 2024 +0100

    adding third.txt
```

## 15- Changez vos paramètres globaux en `core.pager=cat` :

Cela signifie que les résultats des commandes Git qui sont normalement paginés seront désormais affichés directement dans la console sans utiliser de paginador externe ,

```
maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git config --global core.pager cat
```

## 16- Ecrivez la commande appropriée pour lister toutes les configurations globales de git sur votre machine :

Pour lister toutes les configurations , on utilise la commande:  
< git config --global --list > ,

```

maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ git config --global --list
user.name=Maria Chiheb
user.email=airamchiheb@gmail.com
core.pager=cat

maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)

```

## 17- git config --global :

la commande < git config --global > nous aide pour savoir comment utiliser les commandes de < git config > ,

```

maria@DESKTOP-MK89IH8 MINGW64 ~/learn_git (master)
$ 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

```

