# TP

## GIT

Stcherbinine Mattéo | 27/02/23

# Introduction

Dans ce TP nous allons découvrir GIT.

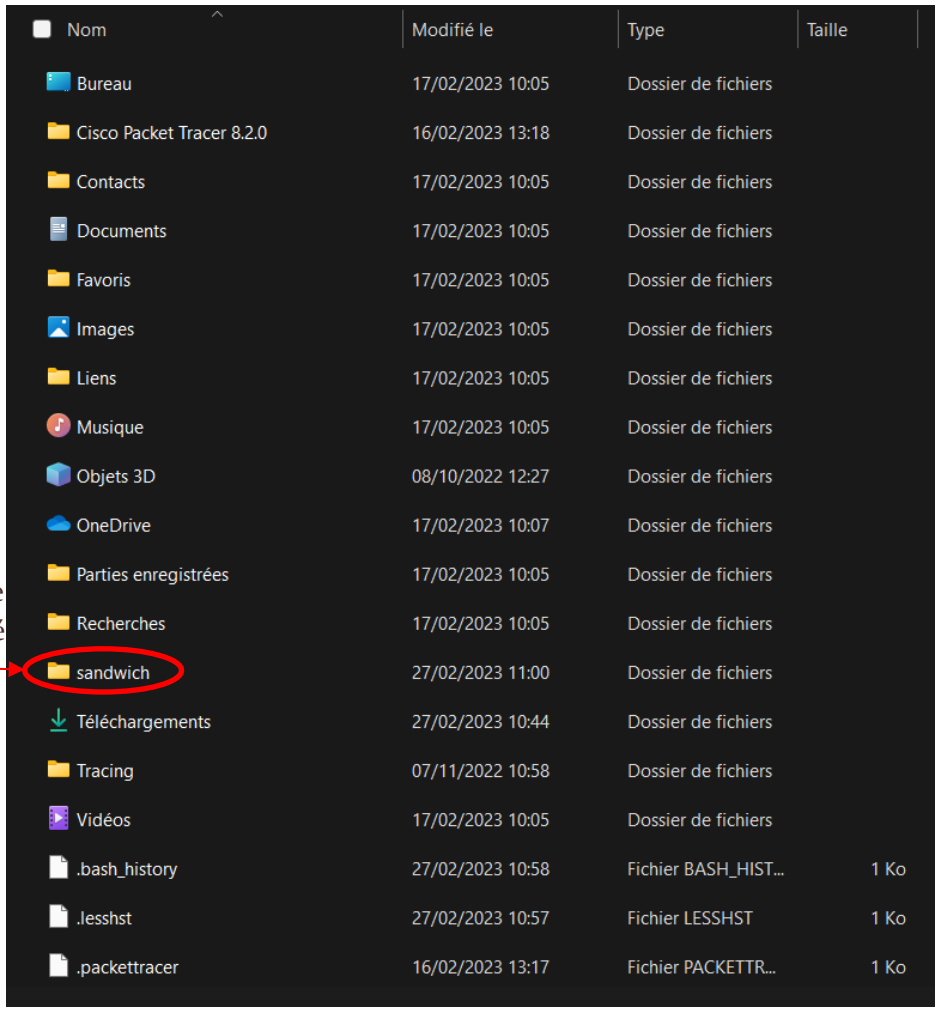
Prérequis :

- Installer GIT

# Première étape :

Nous allons lancer GIT (CMD ou Bash peu importe), puis nous allons taper la commande « git init sandwich ». Cela va créer le répertoire sandwich.

```
Exta@DESKTOP-OOVKIQD MINGW64 ~ (master)
$ git init sandwich
Initialized empty Git repository in C:/Users/Exta/sandwich/.git/
```

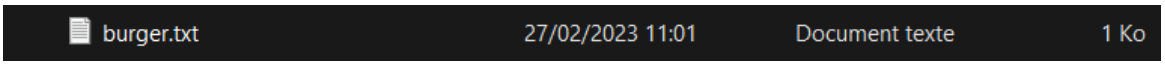


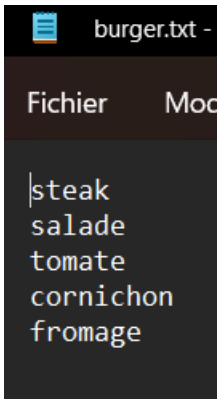
| Nom | Modifié le | Type | Taille |
|---|---|---|---|
| Bureau | 17/02/2023 10:05 | Dossier de fichiers | |
| Cisco Packet Tracer 8.2.0 | 16/02/2023 13:18 | Dossier de fichiers | |
| Contacts | 17/02/2023 10:05 | Dossier de fichiers | |
| Documents | 17/02/2023 10:05 | Dossier de fichiers | |
| Favoris | 17/02/2023 10:05 | Dossier de fichiers | |
| Images | 17/02/2023 10:05 | Dossier de fichiers | |
| Liens | 17/02/2023 10:05 | Dossier de fichiers | |
| Musique | 17/02/2023 10:05 | Dossier de fichiers | |
| Objets 3D | 08/10/2022 12:27 | Dossier de fichiers | |
| OneDrive | 17/02/2023 10:07 | Dossier de fichiers | |
| Parties enregistrées | 17/02/2023 10:05 | Dossier de fichiers | |
| Recherches | 17/02/2023 10:05 | Dossier de fichiers | |
| sandwich | 27/02/2023 11:00 | Dossier de fichiers | |
| Téléchargements | 27/02/2023 10:44 | Dossier de fichiers | |
| Tracing | 07/11/2022 10:58 | Dossier de fichiers | |
| Vidéos | 17/02/2023 10:05 | Dossier de fichiers | |
| .bash_history | 27/02/2023 10:58 | Fichier BASH_HIST... | 1 Ko |
| .lesshst | 27/02/2023 10:57 | Fichier LESSHST | 1 Ko |
| .packettracer | 16/02/2023 13:17 | Fichier PACKETTR... | 1 Ko |

On a bien le répertoire sandwich qui s'est créé ici

## Deuxième étape :

Nous allons créer un fichier texte dans ce répertoire :




Puis dans ce fichier texte nous allons écrire du contenu.

## Troisième étape :

Nous allons ajouter le burger.txt avec la commande git add burger.txt :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~ (master)
$ git add burger.txt
fatal: pathspec 'burger.txt' did not match any files
```

Ce message d'erreur apparaît car nous ne sommes pas dans le bon fichier. Pour se faire on doit se diriger dans le dossier sandwich avec la commande : cd sandwich. On peut maintenant faire notre : git add burger.txt

Maintenant, vérifions la modification du fichier avec la commande git status :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   burger.txt
```

On va utiliser la commande : git diff –-cached

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git diff --cached
diff --git a/burger.txt b/burger.txt
new file mode 100644
index 0000000..3788869
--- /dev/null
+++ b/burger.txt
@@ -0,0 +1,5 @@
+steak
+salade
+tomate
+cornichon
+fromage
```

On voit bien le contenu du fichier texte.

## Quatrième étape :

Maintenant on va Commiter une modification. On utiliser la commande :  git commit -m "<votre_exemple>". Dans notre cas on va rester sur les burgers pour l'exemple :

git commit -m "Burger sans crudités"

Voici le message d'erreur. Git veut savoir qui on est pour qu'il garde une trace de qui a fait quoi. Dans notre cas le nom et l'email importe peu mais dans un cas réel il faudra s'identifier correctement.

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git commit -m "Burger sans crudités"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Exta@DESKTOP-OOVKIQD.(none)')

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$
```

Voici les deux commandes utilisées :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git config --global user.email "burger@sauceketchup.com"

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git config --global user.name "Miam"
```

Une fois que Git nous connaît on peut retaper la commande :

git commit -m "Burger sans crudités"

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git commit -m "Burger sans crudités"
[master (root-commit) 6708ba1] Burger sans crudités
 1 file changed, 5 insertions(+)
 create mode 100644 burger.txt
```

Une fois la commande effectuée on vérifier la modification avec : get status

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Une fois le statut contrôlé on va vérifier les log pour voir qui a modifié quoi, pour se faire on va utiliser la commande git log :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git log
commit 6708ba181ab81edad35e66ce446398e27dd2ec19 (HEAD -> master)
Author: Miam <burger@sauceketchup.com>
Date:   Mon Feb 27 11:38:17 2023 +0100

    Burger sans crudités
```

On peut voir que c'est l'utilisateur Miam qui a modifié le lundi 27 Février 2023 à 11h38

La modification était : Burger sans crudités.

## Cinquième étape :

Nous allons créer deux autres fichiers textes portant des noms de sandwich :

| | | | |
|---|---|---|---|
| burger.txt | 27/02/2023 11:01 | Document texte | 1 Ko |
| hot_dog.txt | 27/02/2023 11:59 | Document texte | 1 Ko |
| jambon_beurre.txt | 27/02/2023 11:59 | Document texte | 1 Ko |

jambon_beurre.txt - B...

Fichier Modifier Affichage

```
pain
beurre
jambon
```

hot_dog.txt - Bloc-notes

Fichier Modifier Affichage

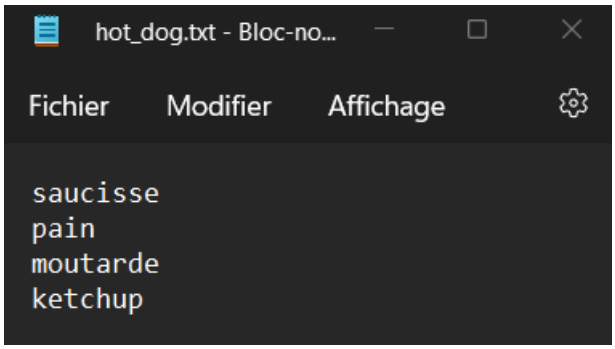
```
pain
saucisse
ketchup
moutarde
```

```
diff --git a/jambon_beurre.txt b/jambon_beurre.txt
index e5e0942..e90183f 100644
--- a/jambon_beurre.txt
+++ b/jambon_beurre.txt
@@ -1,3 +1,3 @@
-pain
-beurre
-jambon
+beurre
+pain
+jambon
\ No newline at end of file

diff --git a/jambon_beurre.txt b/jambon_beurre.txt
new file mode 100644
index 0000000..e5e0942
--- /dev/null
+++ b/jambon_beurre.txt
@@ -0,0 +1,3 @@
+pain
+beurre
+jambon
```

On voit bien le contenu des fichiers textes qu'on vient d'ajouter.

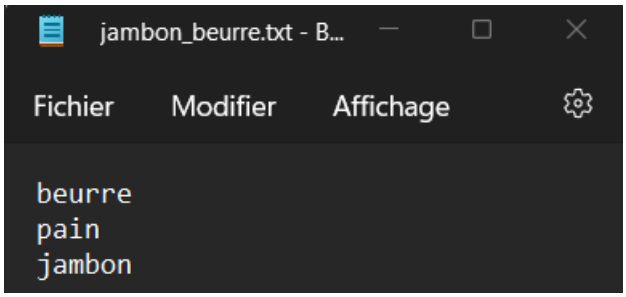
Maintenant on va effectuer des modifications sur les fichiers puis tester si on voit ces modifications.

On modifie l'ordre de notre contenu dans le fichier hot dog par exemple :



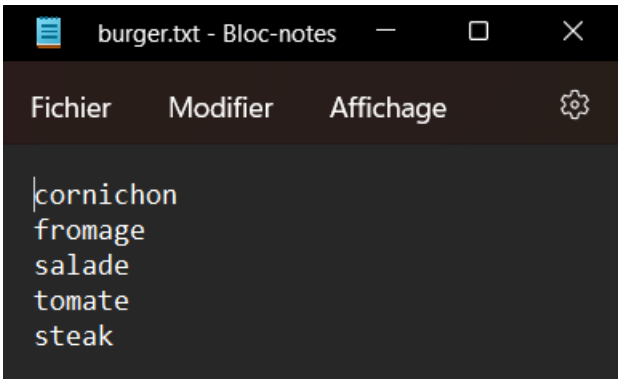

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git diff
diff --git a/hot_dog.txt b/hot_dog.txt
index e4cf0fe..297162c 100644
--- a/hot_dog.txt
+++ b/hot_dog.txt
@@ -1,4 +1,4 @@
-pain
-saucisse
-ketchup
-moutarde
\ No newline at end of file
+saucisse
+pain
+moutarde
+ketchup
\ No newline at end of file

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$
```

On fait pareil pour le fichier jambon beurre :




```
diff --git a/jambon_beurre.txt b/jambon_beurre.txt
index e5e0942..e90183f 100644
--- a/jambon_beurre.txt
+++ b/jambon_beurre.txt
@@ -1,3 +1,3 @@
-pain
-beurre
-jambon
+beurre
+pain
+jambon
\ No newline at end of file
```

Ensuite on peut faire pareil pour le burger :




```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git diff
diff --git a/burger.txt b/burger.txt
index 3788869..544c4e3 100644
--- a/burger.txt
+++ b/burger.txt
@@ -1,5 +1,5 @@
-steak
-salade
-tomate
 cornichon
 fromage
+salade
+tomate
+steak
\ No newline at end of file
```

Si on remodifie chaque fichier pour laisser que la viande par exemple :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git diff
diff --git a/burger.txt b/burger.txt
index 3788869..d0492cd 100644
--- a/burger.txt
+++ b/burger.txt
@@ -1,5 +1 @@
-steak
-salade
-tomate
-cornichon
-fromage
+steak
\ No newline at end of file
diff --git a/hot_dog.txt b/hot_dog.txt
index e4cf0fe..3728b6d 100644
--- a/hot_dog.txt
+++ b/hot_dog.txt
@@ -1,4 +1 @@
-pain
-saucisse
-ketchup
-moutarde
\ No newline at end of file
+saucisse
\ No newline at end of file
diff --git a/jambon_beurre.txt b/jambon_beurre.txt
index e5e0942..b88dc6b 100644
--- a/jambon_beurre.txt
+++ b/jambon_beurre.txt
@@ -1,3 +1 @@
-pain
-beurre
-jambon
+jambon
\ No newline at end of file
```

On voit bien qu'il reste que les viandes

Ensuite on va modifier notre fichier l'ajouter mais ne pas le commit :

On vérifie avec le git status nos modifications

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git add burger.txt

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git add hot_dog.txt

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   burger.txt
        modified:   hot_dog.txt
```

Ensuite on va utiliser la commande : get reset qui permet de revenir au dernier commit fait sur le ou les fichier(s) en question :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git reset burger.txt
Unstaged changes after reset:
M       burger.txt
```

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git reset hot_dog.txt
Unstaged changes after reset:
M       burger.txt
M       hot_dog.txt
```

Une fois le reset fais on peut faire un git checkout :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git checkout burger.txt
Updated 1 path from the index

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git checkout hot_dog.txt
Updated 1 path from the index

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .txt

nothing added to commit but untracked files present (use "git add" to track)

Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$
```

Ensuite on va regarder l'historique de notre dépôt avec la commande git log :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich (master)
$ git log
commit 1b35b21ebcc813754b5135d23edb7f92bf6d8f27 (HEAD -> master)
Author: Miam <burger@sauceketchup.com>
Date:   Mon Mar 13 08:58:26 2023 +0100

    Modif burger et hotdog

commit 6708ba181ab81edad35e66ce446398e27dd2ec19
Author: Miam <burger@sauceketchup.com>
Date:   Mon Feb 27 11:38:17 2023 +0100

    Burger sans crudités
```

Ensuite on prend le commit du 27 Février :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich ((1b35b21...))
$ git checkout 6708ba181ab81edad35e66ce446398e27dd2ec19
Previous HEAD position was 1b35b21 Modif burger et hotdog
HEAD is now at 6708ba1 Burger sans crudités
```

Nous sommes maintenant revenus à la version du 27 février. Si on veut annuler la commande et revenir où l'on en était on peut faire la commande git checkout master :

```
Exta@DESKTOP-OOVKIQD MINGW64 ~/sandwich ((6708ba1...))
$ git checkout master
Previous HEAD position was 6708ba1 Burger sans crudités
Switched to branch 'master'
```