



UNIVERSITÉ
DE TECHNOLOGIE D'HAÏTI
Faculté des Sciences Informatiques



Université de Technologie d'Haïti

UNITECH

Diplôme d'Étude Supérieure Spécialisé (DESS) Technologie de l'Information et l'Informatique (TI)

Option : Administration De Réseaux
Cours de : Gestion Des Risques Informations

Professeur : Ing.Austin Wafo PhD

Préparé par : Guytompous Junior Desmoulin

Session-I Mars-Mai 2025

TRAVAUX PRATIQUES Git, GitHub

Exercice 1 : Initialiser un dépôt Git local

Objectif : Créer un projet versionné localement.

1. Crée un dossier **mon-projet-git**

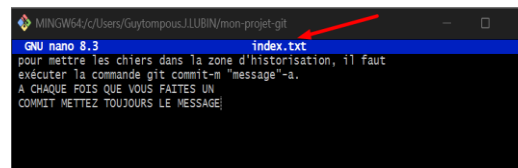
```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~ (master)
$ mkdir mon-projet-git

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~ (master)
$ |
```

2. Crée un fichier **index.txt** avec une phrase

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~ (master)
$ touch index.txt

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~ (master)
$ mv index.txt mon-projet-git/
```



```
GNU nano 8.3 index.txt
pour mettre les chiens dans la zone d'historisation, il faut
exécuter la commande git commit-m "message"-a.
A CHAQUE FOIS QUE VOUS FAITES UN
COMMIT METTEZ TOUJOURS LE MESSAGE
```

3. Initialise Git et fais un premier **commit**

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git init
Initialized empty Git repository in C:/Users/Guytompous.J.LUBIN/mon-projet-git/.git/

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git add index.txt
warning: in the working copy of 'index.txt', LF will be replaced by CRLF the next ti
me Git touches it

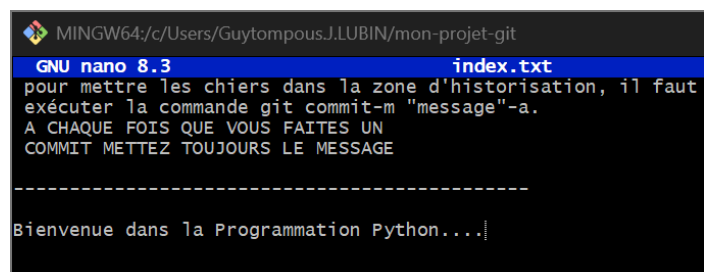
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git commit -m "Travaux Pratique"
[master (root-commit) 1da1833] Travaux Pratique
1 file changed, 4 insertions(+)
create mode 100644 index.txt

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ |
```

Exercice 2 : Travailler avec l'historique Git

Objectif : Modifier un fichier, suivre et annuler les changements.

1. Modifie **index.txt**



```
MINGW64/c/Users/Guytompous.J.LUBIN/mon-projet-git
GNU nano 8.3 index.txt
pour mettre les chiens dans la zone d'historisation, il faut
exécuter la commande git commit-m "message"-a.
A CHAQUE FOIS QUE VOUS FAITES UN
COMMIT METTEZ TOUJOURS LE MESSAGE

-----

Bienvenue dans la Programmation Python....|
```

2. Vérifie les changements

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index.txt

no changes added to commit (use "git add" and/or "git commit -a")

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git diff
warning: in the working copy of 'index.txt', LF will be replaced by CRLF the next time Git touches it
diff --git a/index.txt b/index.txt
index e7ec642..46d2b94 100644
--- a/index.txt
+++ b/index.txt
@@ -2,3 +2,7 @@
 exécuter la commande git commit-m "message"-a.
 A CHAQUE FOIS QUE VOUS FAITES UN
 COMMIT METTEZ TOUJOURS LE MESSAGE

-Bienvenue dans la Programmation Python....
```

3. Annule les modifications

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git checkout -- index.txt
```

Exercice 3 : Ajouter un fichier README.md

Objectif : Créer un README simple et le commiter.

1. Crée un fichier **README.md** avec :

md CopierModifier

Mon Projet Git

Petit test de versionnage avec Git.

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ echo "#Mon Projet Git\n Petit test de versionnage avec Git." > README.md
```

2. Ajoute et committe le fichier

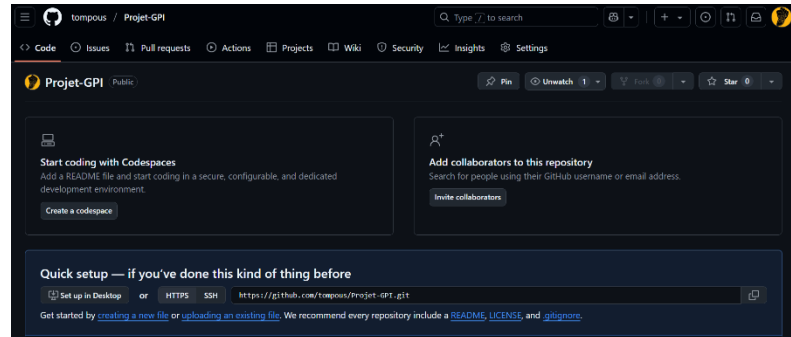
```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git commit -m "On ajoute le Fichier README.md"
[master 00d1c6b] On ajoute le Fichier README.md
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Exercice 4 : Créer un dépôt GitHub et pousser le projet

Objectif : Mettre ton projet en ligne.

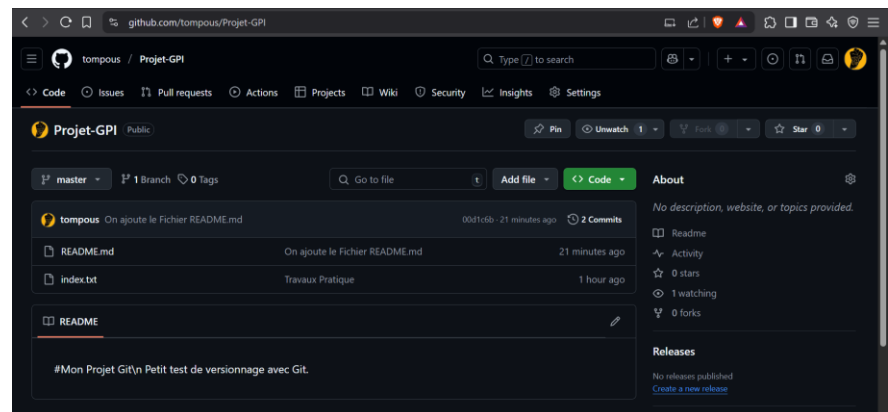
1. Crée un nouveau repo sur GitHub sans README



2. Ajoute le remote et pousse ton projet

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git remote add origin https://github.com/tompous/Projet-GPI.git

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 706 bytes | 706.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/tompous/Projet-GPI.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```



Exercice 5 : Créer une nouvelle branche

Objectif : Expérimenter une fonctionnalité sans casser le code principal.

1. Crée une branche **nouvelle-fonction**

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git checkout -b nouvelle-fonction
Switched to a new branch 'nouvelle-fonction'
```

2. Ajoute un fichier fonction.py

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git checkout -b nouvelle-fonction
Switched to a new branch 'nouvelle-fonction'

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ touch fonction.py

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ ls
README.md  fonction.py  index.txt

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git add fonction.py

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git commit -m "Ajout de nouvelle-fonction" -a
[nouvelle-fonction d760855] Ajout de nouvelle-fonction
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fonction.py

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
```

3. Merge cette branche dans main

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git merge nouvelle-fonction
Updating 00d1c6b..d760855
Fast-forward
 fonction.py | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fonction.py
```

Exercice 6 : Gérer un conflit

Objectif : Apprendre à résoudre un conflit Git.

1. Sur main, ajoute une ligne à index.txt

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (Nouvelle-Fonction)
$ echo "Salut tout le monde!" > index.txt

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (Nouvelle-Fonction)
$ git add index.txt
warning: in the working copy of 'index.txt', LF will be replaced by CRLF the next time Git touches it

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (Nouvelle-Fonction)
$ git commit -m "Modification Index.txt" -a
[Nouvelle-Fonction 7d93fe1] Modification Index.txt
1 file changed, 1 insertion(+), 1 deletion(-)
```

2. Sur une autre branche **update-index**, modifie la même ligne

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (Nouvelle-Fonction)
$ git checkout -b Update-Index
Switched to a new branch 'Update-Index'

Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (Update-Index)
$ git branch
  Nouvelle-Fonction
* Update-Index
  master
```

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ git merge Nouvelle-Fonction
Auto-merging index.txt
CONFLICT (content): Merge conflict in index.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Exercice 7 : Améliorer le README.md

Objectif : Rédiger un README plus professionnel.

Ajoute les sections suivantes :

- Description
- Installation
- Utilisation
- Auteur

```
Guytompous.J.LUBIN@DESKTOP-DHOIARP MINGW64 ~/mon-projet-git (master)
$ echo "Ce projet vise à mettre en pratique les concepts de base et avancés de Git et GitHub. Git et GitHub sont un espace ou un environnement qui emballe les fées de test personnel dans lequel je peux explorer les différentes commandes liées aux versions, tout en suivant leurs meilleures pratiques en développement collaboratif. L'accent est mis sur l'apprentissage progressif dans des cas concrets tels que la création de succursales, les fusions ou la gestion des conflits.

L'installation est simple: il suffit de cloner le dépôt à l'aide de la commande git clone https://github.com/tompous/Projet-GPI.git. Aucune configuration supplémentaire n'est requise, vous permettant de commencer immédiatement à utiliser Git sans avoir besoin de configurer de dépendances ou de paramètres techniques.

L'utilisation des commandes exécuter dans ce projet se concentre principalement dans le cadre d'un test. Nous permettons de mieux sauvegarder nos projets et documents dans un espace dédié aux professionnelles éclairer et aviser, prenons le cas du fichier 'index.txt' nous permet pour effectuer diverses modifications, ajout, commit, tester des commandes comme 'git add', 'git commit', 'echo' etc. Le fichier README.md subit plusieurs modifications au fil des exercices pour documenter les étapes franchies et les notions apprises, créant ainsi une trace claire de l'évolution du projet.

Ce travaux pratique sur l'utilisation et l'initialisation de git et GitHub m'a appris s Guytompous J. DESMOULIN, à des fins éducatives et professionnelle. Et, j'ai appris aussi de l'utilisation et de l'importance de git GitHub dans la vie professionnel et éducative d'une personne. N'hésitez pas à proposer des suggestions ou poser des questions en ouvrant une issue. Pour tout contact direct, veuillez me contacter a l'adresse mail : guytompousjuniordesmoulin@gmail.com" > README.md
bash: README.md: command not found
```



