



**UNIVERSITÉ
DE TECHNOLOGIE D'HAÏTI**

53 et 57, Ave N, Turgeau, Port-au-Prince, Haïti

Email : info@unitech.edu.ht

Faculté des Sciences, de Génie et d'Architecture

Programme : Diplôme de 2^{ème} cycle en technologie de l'information

Cours : Gestion des Risques Informatiques

Objet : Devoir (Travaux Pratiques Git, GitHub)

Professeur :

Dr.-Ing. Austin Waffo Kouhoué, PhD

Présenté par :

Jonel LUBIN

Soumis le, 10 – 05– 2025

Exercice 1 : Initialiser un dépôt Git local

Objectif : Créer un projet versionné localement

1. Création d'un dossier **mon-projet-git**

Après avoir lancé git : voici le script de cette question :

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (master)
$ mkdir mon-projet-git
```

2. Création d'un fichier index.txt avec une phrase

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ touch index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ nano index.txt
```

J'ouvre le fichier avec nano index.txt puis Ctrl +x , puis y suivi de Enter

```
GNU nano 8.3 index.txt
ceci est mon premier fichier versionné avec git|
```

3. Initialise Git et fais un premier commit

```
JLMsc@DESKTOP-KHTOMKE 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 time Git touches it
```

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ git commit -m "Premier commit : ajout du fichier index.txt"
[master (root-commit) e9bbdb7] Premier commit : ajout du fichier index.txt
1 file changed, 1 insertion(+)
create mode 100644 index.txt
```

Exercice 2 : Travailler avec l'historique Git

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

1. Modifions l'index

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ nano index.txt
```

```
MINGW64:/c/Users/JLMsc/mon-projet-git
GNU nano 8.3 index.txt
ceci est mon premier fichier versionné avec git
je modifie mon premier fichier|
```

2. Vérifions les changements

a) Voyons l'état du dépôt

```
$ 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")
```

b) Affichons les différences avec le dernier commit :

```
$ 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 183b5fb..0ebb1c3 100644
--- a/index.txt
+++ b/index.txt
@@ -1,3 @@
- ceci est mon premier fichier versionné avec git
+
+ je modifie mon premier fichier
```

3. Annulons les modifications

```
$ git restore index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Exercice 3 : Ajouter un fichier README.md Objectif :

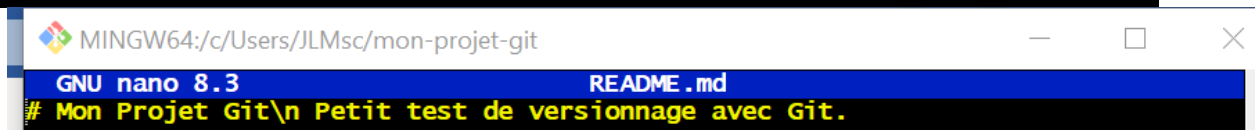
Créer un README simple et le commiter.

1. Crée un fichier README.md avec :

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ touch README.md

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ ls
README.md  index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ nano README.md
```



2. Ajoute et committe le fichier

```
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time git touches it

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ ls
README.md  index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ git commit -m "Ajout du fichier README.md"
[master dd483a8] Ajout du 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.

Énoncé :

1. Crée un nouveau repo sur GitHub sans README


Pour cette étape je lance <https://github.com/>, j'ai déjà un compte sur github et je crée un new repository, je laisse la description publique avec une description : « mon premier projet » puis je clique sur Create repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*

Owner *

 JLubin-prog

Repository name *

mon-projet-git


 mon-projet-git is available.

Great repository names are short and memorable. Need inspiration? How about [ideal-octo-chainsaw](#) ?

Description (optional)

mon premier projet

☒ Public

 Private

Anyone on the internet can see this repository. You choose who can commit.



mon-projet-git

Public

2. Ajoute le remote et pousse ton projet

Je copie l'Url depuis le github puis coller dans git

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ git remote add origin https://github.com/JLubin-prog/mon-projet-git.git

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (master)
$ git branch -M main
```

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git puush -u origin main
git: 'puush' is not a git command. See 'git --help'.

The most similar command is
    push

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 591 bytes | 295.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JLubin-prog/mon-projet-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

Voici le résultat sur github

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'main' selected, '1 Branch', '0 Tags', a search bar, and a 'Code' button. Below this, the repository name 'mon premier projet' is displayed. The commit history table shows two commits: 'JLubin Ajout du fichier README.md' (28 minutes ago, 2 commits) and 'Premier commit : ajout du fichier index.txt' (53 minutes ago). The README file is selected and open, displaying the text 'Mon Projet Git\n Petit test de versionnage avec Git.'.

Exercice 5 : Créer une nouvelle branche Objective :

Expérimenter une fonctionnalité sans casser le code principal. Énoncé :

1. Crée une branche nouvelle-fonction

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git branch nouvelle-fonction

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

Basculons sur cette branche avec la commande git Checkout nouvelle-fonction

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git checkout nouvelle-fonction
Switched to branch 'nouvelle-fonction'

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ |
```

2. Ajout d'un fichier fonction.py

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ touch fonction.py

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ nano fonction.py
```

Il faut faire le transfert vers le dépôt distant

```
README.md  fonction.py  index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git push -u origin nouvelle-fonction
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nouvelle-fonction' on GitHub by visiting:
remote:   https://github.com/JLubin-prog/mon-projet-git/pull/new/nouvelle-fonction
remote:
To https://github.com/JLubin-prog/mon-projet-git.git
 * [new branch]      nouvelle-fonction -> nouvelle-fonction
branch 'nouvelle-fonction' set up to track 'origin/nouvelle-fonction'.
```

3. Merge cette branche dans main

on doit retourner dans la branche main

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git merge nouvelle-fonction
Updating dd483a8..38fdf61
Fast-forward
 fonction.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fonction.py

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$
```

Voici le résultat sur github.com

The screenshot shows the GitHub repository page for 'mon-projet-git'. The repository is public and has 2 branches and 0 tags. The main branch is selected. The repository description is 'mon premier projet'. The file list shows 'README.md', 'fonction.py', and 'index.txt'. The commit history shows a merge pull request #1 from JLibin-prog/nouvelle-fonction. The repository has 0 stars, 1 watching, and 0 forks.

Exercice 6 : Gérer un conflit Objectif :

Apprendre à résoudre un conflit Git. Énoncé :

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

Comme on est déjà dans git je vais ajouter une ligne à index.txt

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ nano index.txt
```

The screenshot shows the nano text editor window. The title bar indicates the file is 'index.txt' and it has been modified. The content of the file is 'ceci est mon premier fichier versionné avec git' followed by a new line 'Ligne ajoutée|'.


```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git add index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git commit -m "Ajout d'une ligne dans index.txt sur main"
[main bc9e834] Ajout d'une ligne dans index.txt sur main
1 file changed, 2 insertions(+)

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

2- Sur une autre branche update-index, modifie la même ligne
Je crée un conflit git en modifiant la même ligne sur une autre branche

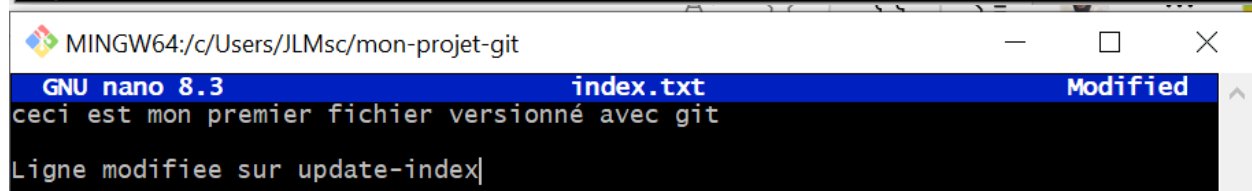
```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git checkout -b update-index
Switched to a new branch 'update-index'

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ |
```

Modifier la même ligne que sur main dans index.txt

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ nano index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ |
```



- Commit la modification

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ git add index.txt

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ git commit -m "modification de la même ligne dans index.txt sur update-index"
[update-index 834e1a3] modification de la même ligne dans index.txt sur update-i
ndex
1 file changed, 1 insertion(+), 1 deletion(-)

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ |
```

- Fusionner avec main pour provoquer un conflit

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (update-index)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

- Lancer la fusion :

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git merge update-index
Updating bc9e834..834e1a3
Fast-forward
 index.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

En résumé, git trouve un conflit dans index.txt, car la même ligne a été modifiée dans deux branches différentes

Pour vérifier le conflit on fait git status

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git satus
git: 'satus' is not a git command. See 'git --help'.

The most similar command is
    status
```



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

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git commit -m "Ajout de la section Description dans ReadMe.md"
[main ffc5fce] Ajout de la section Description dans ReadMe.md
1 file changed, 3 insertions(+)

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

• Installation

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git clone https://github.com/JLubin-prog/mon-projet-git.git
Cloning into 'mon-projet-git'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 0), reused 9 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ cd mon-projet-git

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git/mon-projet-git (main)
$ |
```

• Utilisation

- Nous allons Utiliser

1. git status

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    mon-projet-git/

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

JLMsc@DESKTOP-KHTOMKE MINGW64 ~/mon-projet-git (main)
$ |
```

2. git log

```

Ajout de la section Description dans ReadMe.md

commit 7af5955b9204e4613a4b85808df863a9083b0ad0
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 23:00:40 2025 -0400

Ajour de la section Description dans READ.md

commit 834e1a316029403bd26725e1341a6afb7de17a1 (update-index)
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 22:17:44 2025 -0400

modification de la meme ligne dans index.txt sur update-index

commit bc9e834d1110f526791de65422ae1d9d50d1e213
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 22:06:52 2025 -0400

Ajour d'une ligne dasn index.txt sur main

commit 4a87f548ba63a640920ae73ae0960cf9cf46e8d2 (origin/main)
Merge: dd483a8 38fdf61
Author: JLubin <jone1lubin10@gmail.com>
Date: Fri May 9 21:56:23 2025 -0400

Merge pull request #1 from JLubin-prog/nouvelle-fonction

ajout du fichier fonction.py avec Bonjour

commit 38fdf61d613853dc6ce698b1a0d836a2f657fa3d (origin/nouvelle-fonction, nouvelle-fonction)
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 21:32:36 2025 -0400

ajout du fichier fonction.py avec Bonjour

commit dd483a85bddb3909255a301cb051206b840a31ab
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 20:49:54 2025 -0400

Ajout du fichier README.md

commit e9bbdb7ecc1506ce18d5df6c61fb192e99dc00a9
Author: JLubin <ing.jlubin@gmail.com>
Date: Fri May 9 20:24:59 2025 -0400

```

3. git show

```

JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (master)
$ git show
commit 67a6e9a97959e11fe718172e6b8928f18d9efcd4 (HEAD -> master)
Author: JLubin <ing.jlubin@gmail.com>
Date: Sat May 10 20:25:39 2025 -0400

Initial commit avec README.md

diff --git a/README.md b/README.md
deleted file mode 100644
index 721641a..0000000
--- a/README.md
+++ /dev/null
@@ -1,0,0 @@
-Mon dépôt Git

JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (master)
$ |

```

- Auteur

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (master)
$ nano README.md
```



The screenshot shows a nano editor window titled "MINGW64:/c/Users/JLMsc". The editor is in "GNU nano 8.3" mode, editing the file "README.md". The content of the file is as follows:

```
## Auteur
-Nom ; Jone1 LUBIN
-GitHub : JLUbin-Prog
```

A la fin, j'effectue des commandes comme git add , git commit

```
JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (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

JLMsc@DESKTOP-KHTOMKE MINGW64 ~ (master)
$ git commit -m "Ajout des infos de l'auteur dans le README.md"
[master 2d30b0f] Ajout des infos de l'auteur dans le README.md
1 file changed, 3 insertions(+)
create mode 100644 README.md
```