Misons sur le capital Humain

UNITECH | BRH

Université de Technologie d'Haïti/Banque de la république d'Haïti .

Gestion des risques infomatique

Préparer Par :

- *Johnley-Roosevelt LORVIL*

Le 28/03/2025

Partie 1

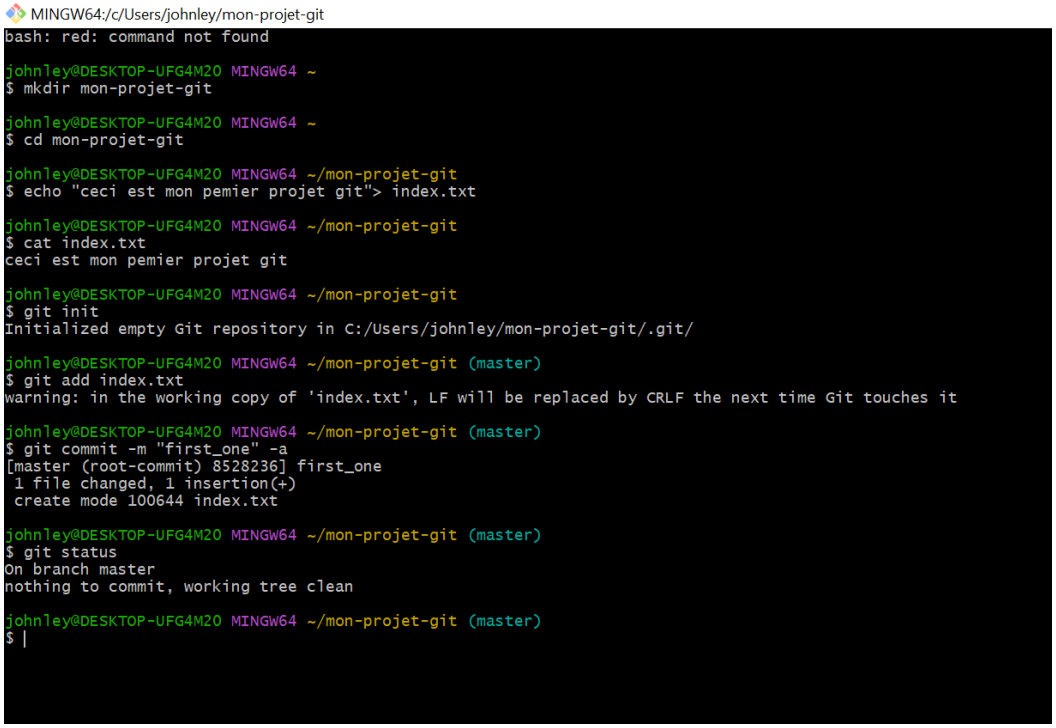
Exercice 1 : Initialiser un dépôt Git local

Objectif : Créer un projet versionné localement.

Énoncé :

1. Crée un dossier mon-projet-git
2. Crée un fichier index.txt avec une phrase
3. Initialise Git et fais un premier commit

L'image que voici est la résolution de l'exercice précédent :



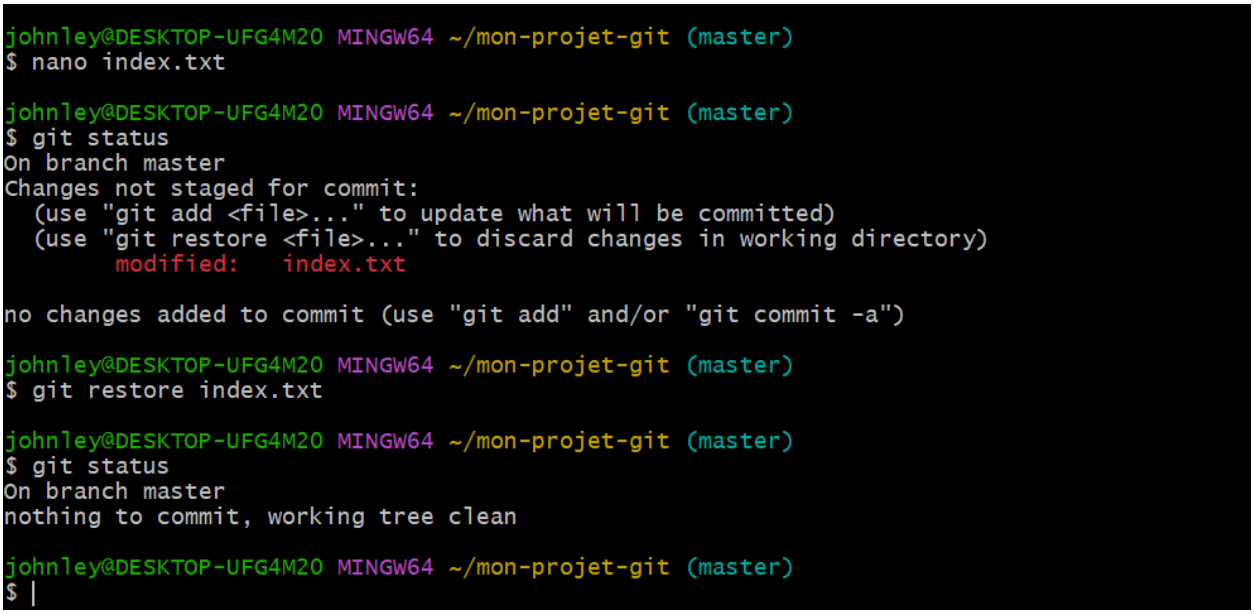
Exercice 2 : Travailler avec l'historique Git

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

Énoncé :

1. Modifie `index.txt`
2. Vérifie les changements
3. Annule les modifications

L'image que voici est la résolution de l'exercice précédent :



Exercice 3 : Ajouter un fichier README.md

Objectif : Créer un README simple et le commiter.

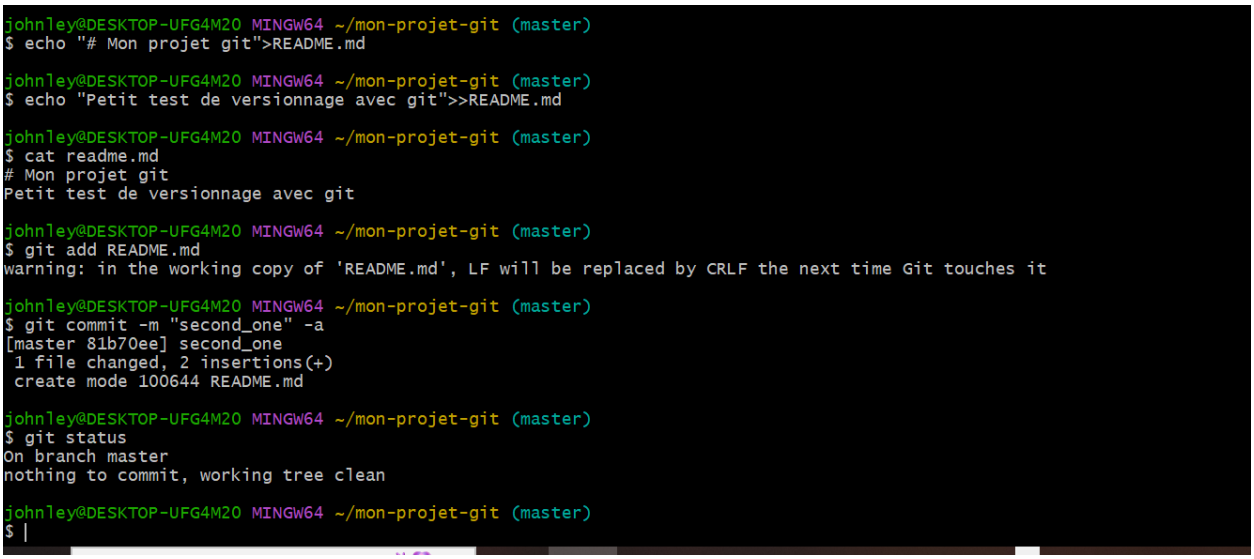
Énoncé :

1. Crée un fichier `README.md` avec :

```
md
CopierModifier
# Mon Projet Git
Petit test de versionnage avec Git.
```

2. Ajoute et committe le fichier

L'image que voici est la résolution de l'exercice précédent :



Exercice 4 : Créer un dépôt GitHub et pousser le projet

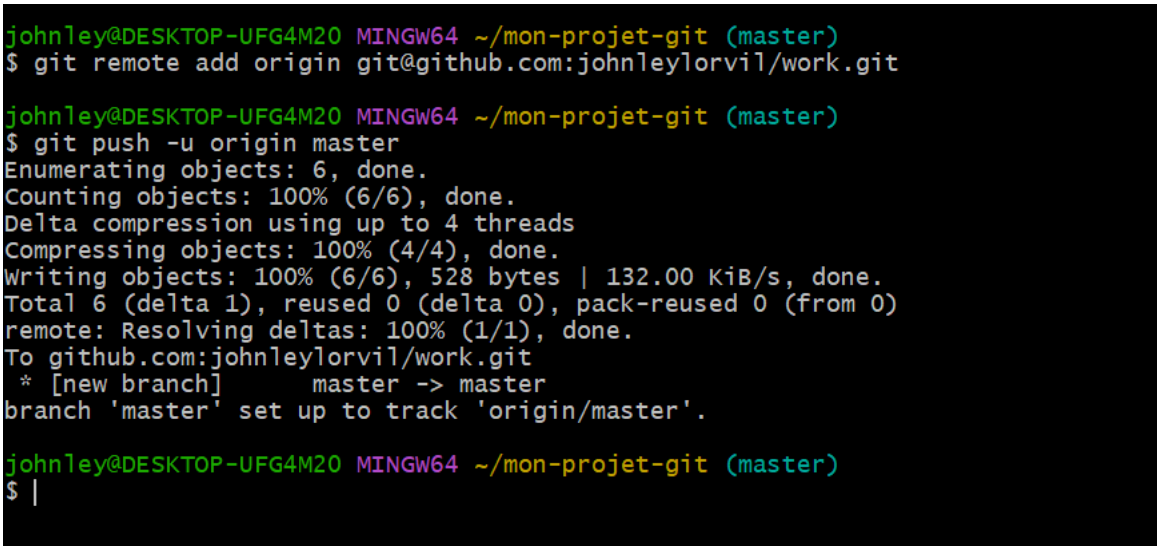
Objectif : Mettre ton projet en ligne.

Énoncé :

1. Crée un nouveau repo sur GitHub sans README

2. Ajoute le remote et pousse ton projet

L'image que voici est la résolution de l'exercice précédent :



Exercice 5 : Créer une nouvelle branche

Objectif : Expérimenter une fonctionnalité sans casser le code principal.

Énoncé :

1. Crée une branche `nouvelle-fonction`
2. Ajoute un fichier `fonction.py`
3. Merge cette branche dans `main`

Les deux images que voici est la résolution de l'exercice précédent :

```
johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (master)
$ git branch
* master

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (master)
$ git branch nouvelle-fonction

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (master)
$ git branch
* master
  nouvelle-fonction

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (master)
$ git branch -M main

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git branch
* main
  nouvelle-fonction

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git checkout nouvelle-fonction
Switched to branch 'nouvelle-fonction'

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ echo "essaie">fonction.py

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ ls
README.md  fonction.py  index.txt

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git add fonction.py
warning: in the working copy of 'fonction.py', LF will be replaced by CRLF the next time Git touches it

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git status
On branch nouvelle-fonction
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   fonction.py
```

```
        new file:   fonction.py


johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git commit -m "troisieme"
[nouvelle-fonction 2fcb27e] troisieme
 1 file changed, 1 insertion(+)
 create mode 100644 fonction.py

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git status
On branch nouvelle-fonction
nothing to commit, working tree clean

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (nouvelle-fonction)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/master'.

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git merge nouvelle-fonction
Updating 81b70ee..2fcb27e
Fast-forward
 fonction.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fonction.py

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ ls
README.md  fonction.py  index.txt

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git push -u origin main
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), 333 bytes | 33.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:      https://github.com/johnleylorvil/work/pull/new/main
remote:
To github.com:johnleylorvil/work.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Exercice 6 : Gérer un conflit

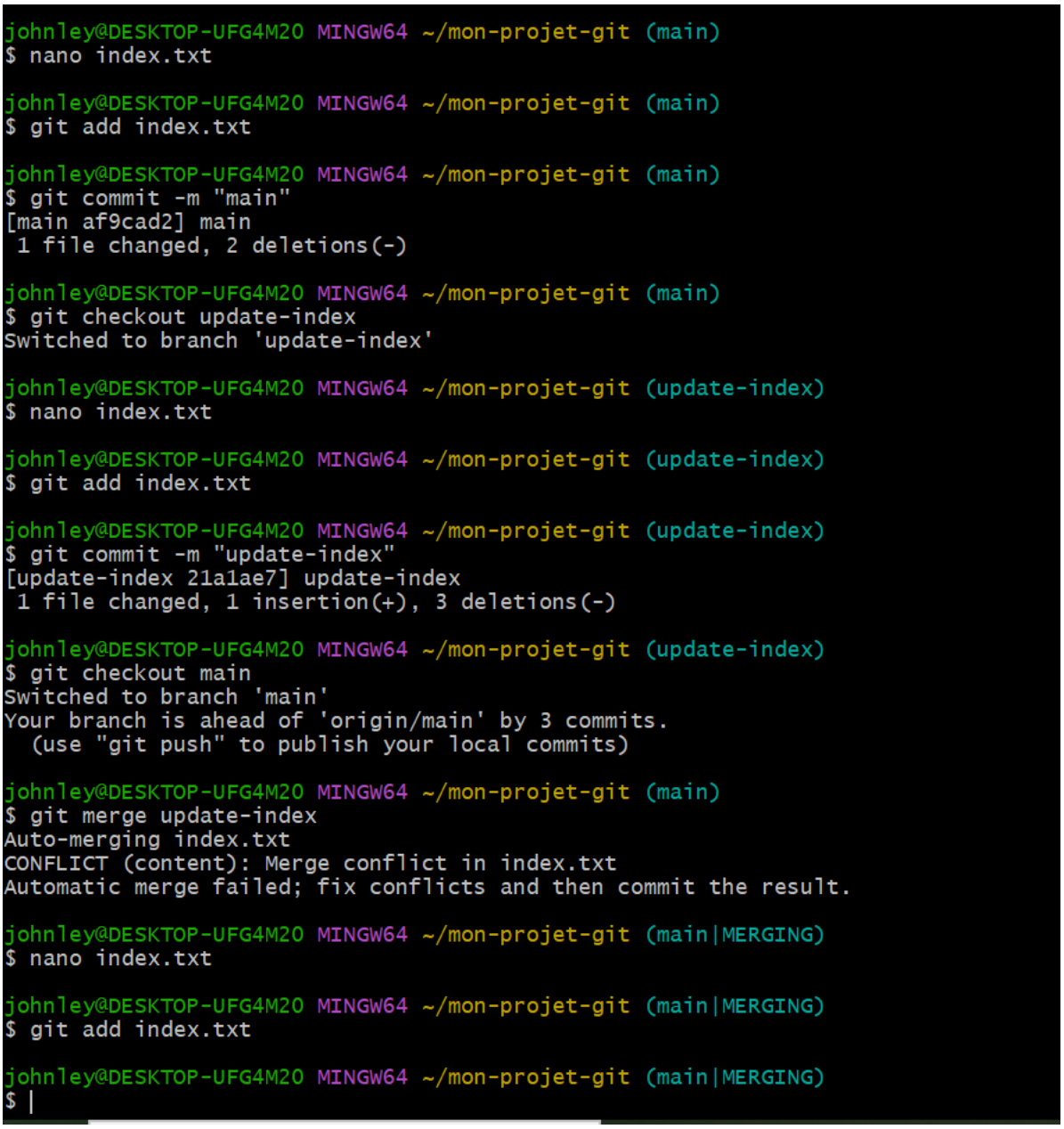
Objectif : Apprendre à résoudre un conflit Git.

Énoncé :

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

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

L'image que voici est la résolution de l'exercice précédent :



Exercice 7 : Améliorer le README.md

Objectif : Rédiger un README plus professionnel.

❖ **Énoncé :**

Ajoute les sections suivantes :

- Description
- Installation
- Utilisation
- Auteur

L'image que voici est la résolution de l'exercice précédent :

```
johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ nano readme.md

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ cat readme.md
# Mon Projet Git

Petit test de versionnage avec Git.

## 📄 Description
Ce projet est un exercice simple pour apprendre à utiliser Git et GitHub.
Il couvre les bases des commandes Git telles que `add`, `commit`, `merge`, `branch` et `push`.

## 🚀 Installation
1. **Clone le dépôt :**
   ```bash
   git clone <URL_DU_REPOSITORY>


johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git add readme.md

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git commit -m "readme_pro" -a
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[main 799d598] readme_pro
 2 files changed, 15 insertions(+), 3 deletions(-)

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$ git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 1.41 KiB | 207.00 KiB/s, done.
Total 13 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:johnleylorvil/work.git
   2fcb27e..799d598  main -> main
branch 'main' set up to track 'origin/main'.

johnley@DESKTOP-UFG4M2O MINGW64 ~/mon-projet-git (main)
$
```
```

Partie 2

Travaux Pratiques GIT et GITHUB

Exercice 1 : Initialiser un dépôt Git

Objectif : Créer un dépôt local et ajouter un fichier

- Créer un dossier ex1_init_repo
- Ajouter un fichier README.md contenant une description

L'image que voici est la résolution de l'exercice précédent :

```
johnley@DESKTOP-UFG4M2O MINGW64 ~
$ mkdir ex1_init_repo

johnley@DESKTOP-UFG4M2O MINGW64 ~
$ cd ex1_init_repo

johnley@DESKTOP-UFG4M2O MINGW64 ~/ex1_init_repo
$ git init
Initialized empty Git repository in C:/Users/johnley/ex1_init_repo/.git/

johnley@DESKTOP-UFG4M2O MINGW64 ~/ex1_init_repo (master)
$ nano readme.md

johnley@DESKTOP-UFG4M2O MINGW64 ~/ex1_init_repo (master)
$ cat readme.md
# Exercice 1 - Initialiser un dépôt Git

Ceci est un projet pour apprendre à initialiser un dépôt Git et ajouter un fichier.

johnley@DESKTOP-UFG4M2O MINGW64 ~/ex1_init_repo (master)
$ ls
readme.md

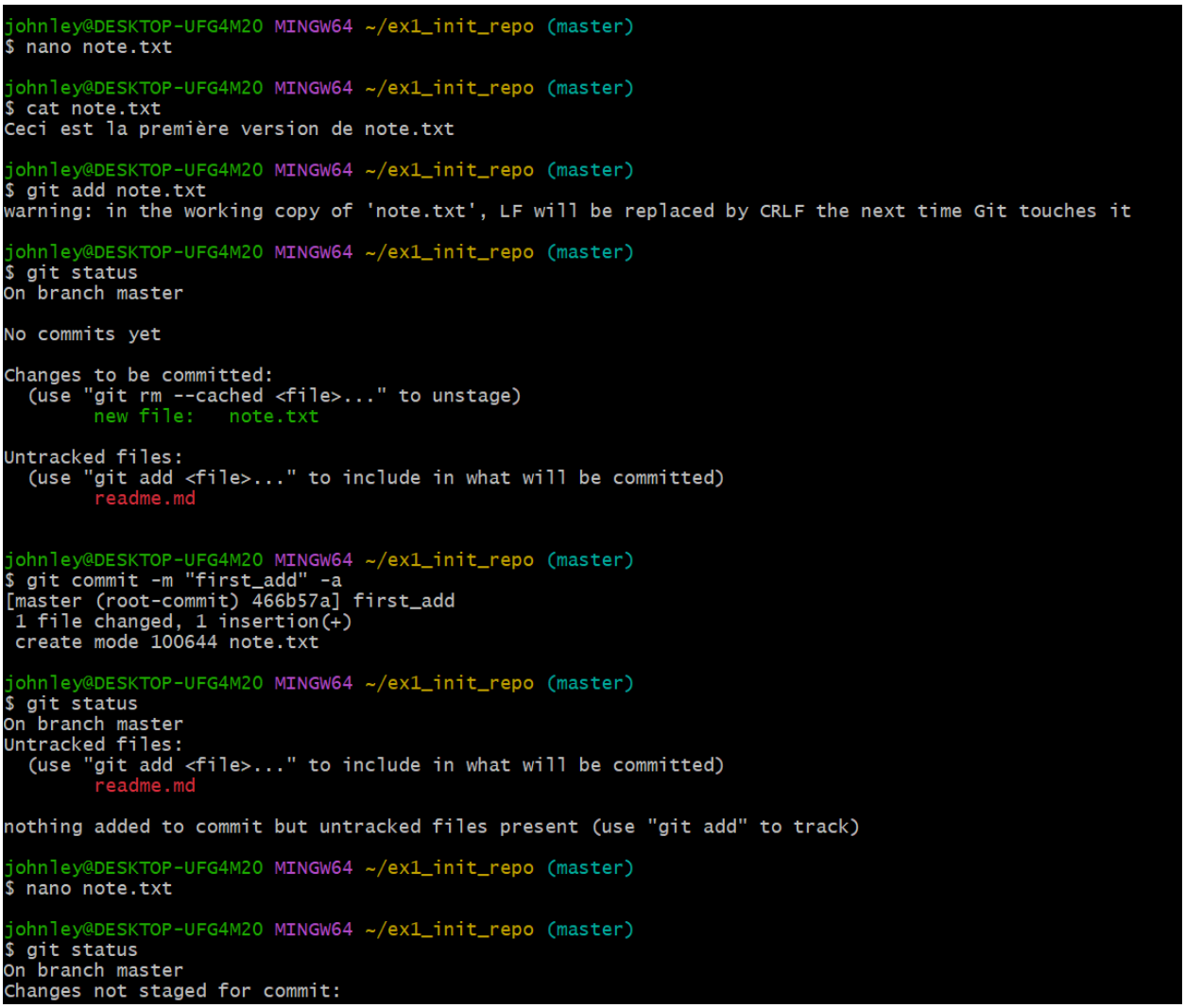
johnley@DESKTOP-UFG4M2O MINGW64 ~/ex1_init_repo (master)
$
```

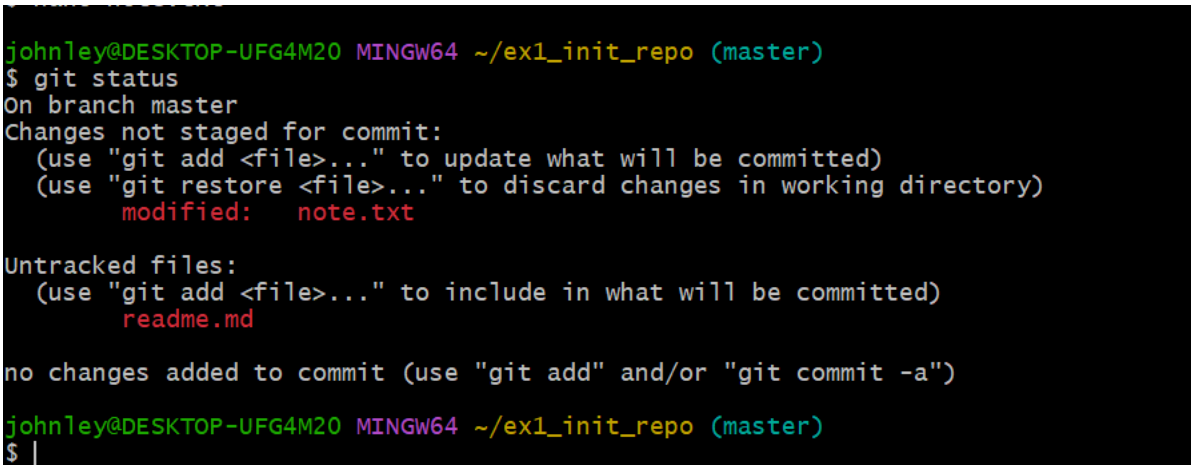
Exercice 2 : Suivre les modifications d'un fichier

Objectif : Observer l'état d'un fichier modifié

 Modifier plusieurs fois note.txt

Les deux images que voici est la résolution de l'exercice précédent :

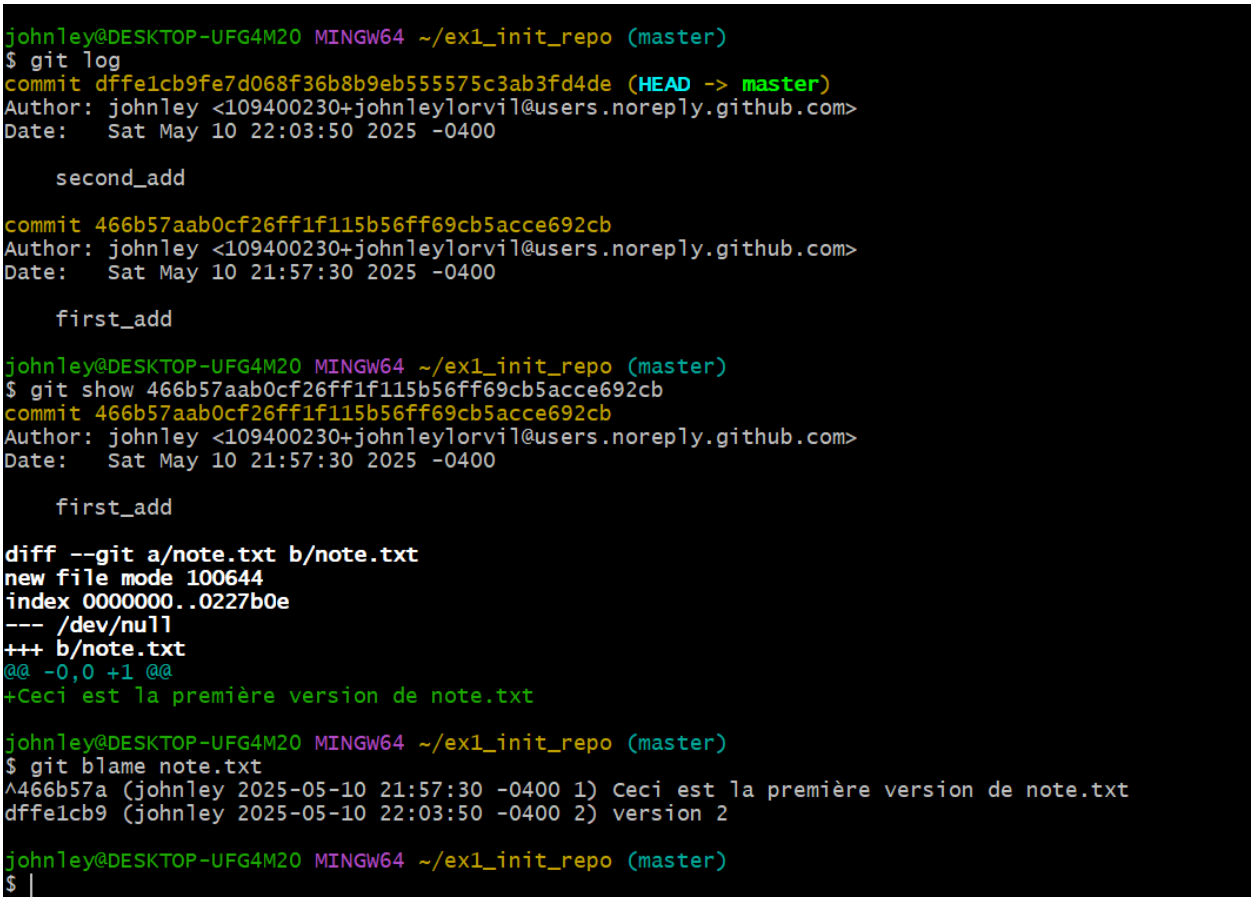




Exercice 3 : Consulter l'historique des changements

Objectif : Utiliser log, show, blame

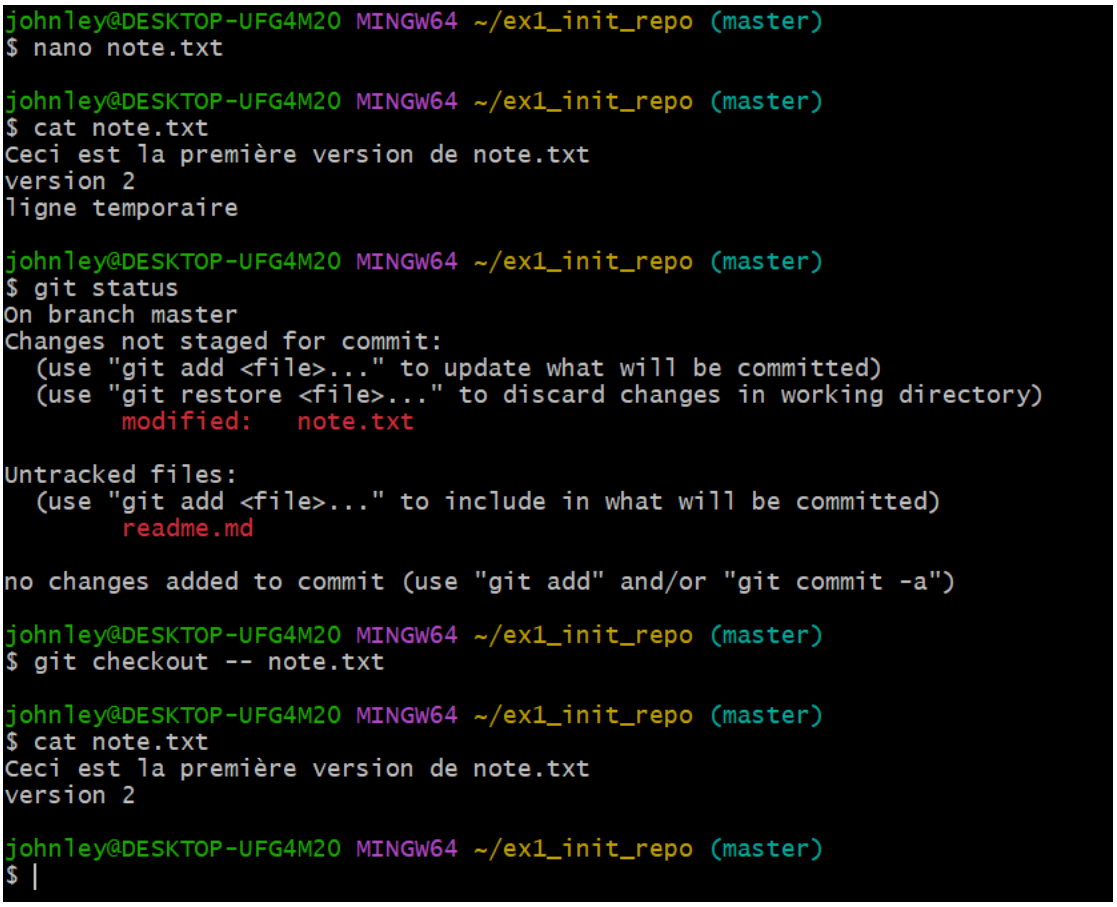
L'image que voici est la résolution de l'exercice précédent :



Exercice 4 : Annuler des changements

Annuler avant le commit

L'image que voici est la résolution de l'exercice précédent :



Exercice 5 : Lier un dépôt local à GitHub

Push vers GitHub

L'image que voici est la résolution de l'exercice précédent :