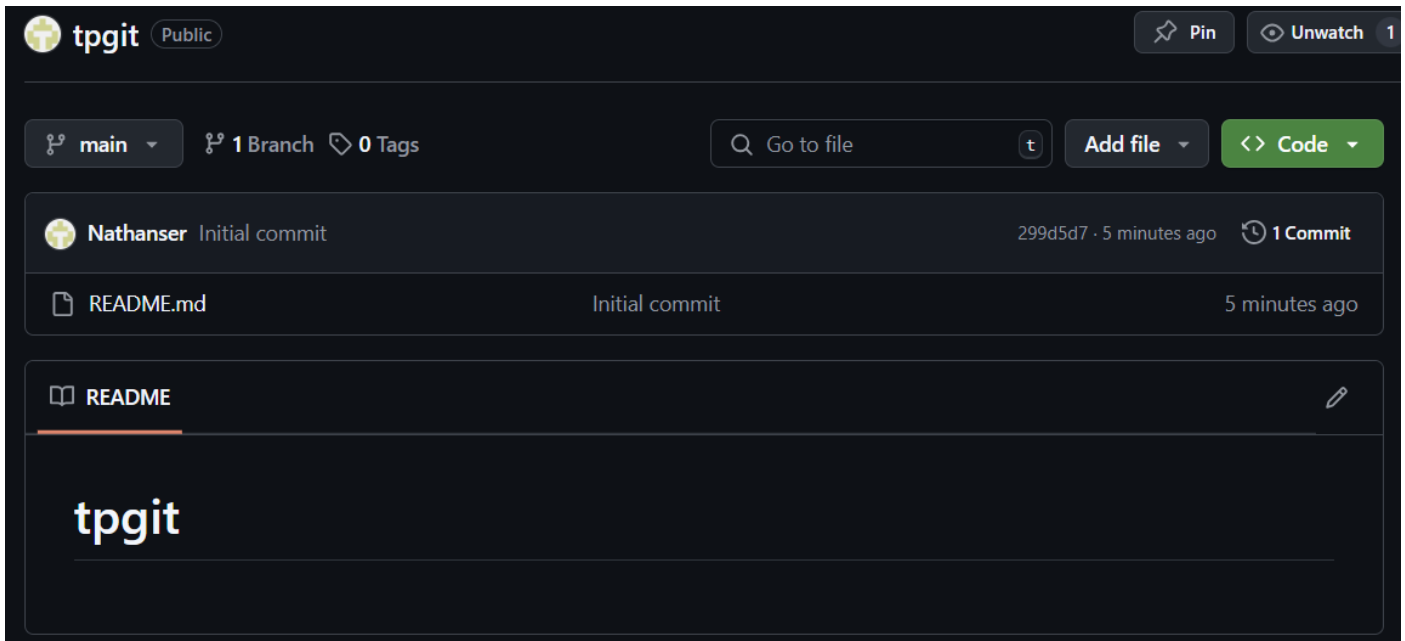


TP 1 Git

1- Création du dépôt distant sur Github :



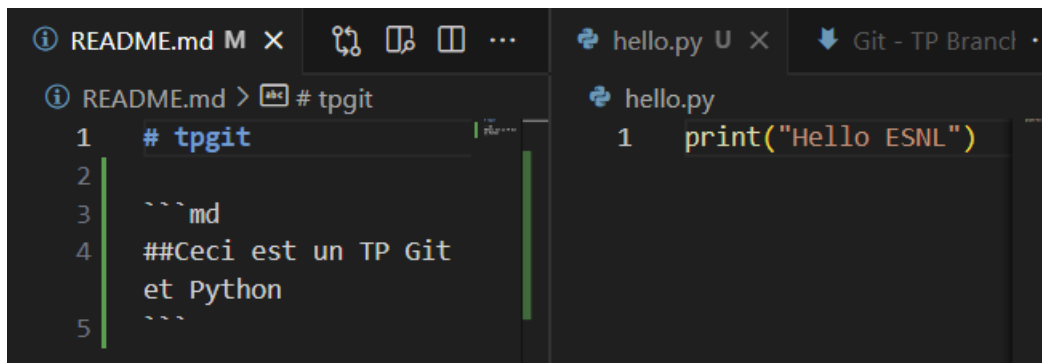
2- Clonage du repository en local :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~  
$ git clone https://github.com/Nathanser/tpgit.git  
Cloning into 'tpgit'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (3/3), done.
```

3- Création d'un dossier tp_git_python + initialisation du dépôt :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)  
$ mkdir tp_git_python_1  
  
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)  
$ cd tp_git_python_1  
  
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit/tp_git_python_1 (main)  
$ git init  
Initialized empty Git repository in C:/Users/natha/tpgit/tp_git_python_1/.git/
```

4- Création du fichier python et readme.md :



The screenshot shows a code editor with two files open. The left file is README.md, containing a header '# tpgit', a code block '```md', and a description '##Ceci est un TP Git et Python' followed by another code block '```'. The right file is hello.py, containing a single line of Python code: 'print("Hello ESNL")'.

5- Statuts du repository affiche que les fichiers ne sont pas sauvegardés dans le dépôt local, ils sont uniquement présent qu'en local :

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.py
    tp_git_python_1/

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

6- Ajout des fichiers sur le dépôt distant et statuts mis à jour :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git add .

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
    new file:   hello.py
```

7- Commit :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git commit -m 'Initialisation du projet'
[main 21c0607] Initialisation du projet
2 files changed, 6 insertions(+), 1 deletion(-)
create mode 100644 hello.py
```

8- Push :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 359 bytes | 359.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Nathanser/tpgit.git
299d5d7..21c0607  main -> main
```

TP 2 Branches

1- Création d'une nouvelle branche « newbranch » :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git branch
* main

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git checkout -b newbranch
Switched to a new branch 'newbranch'

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (newbranch)
$ git branch
main
* newbranch
```

Branches						New branch
Overview Yours Active Stale All						
Q Search branches...						
Branch	Updated	Check status	Behind	Ahead	Pull request	
main	9 minutes ago			Default		...
newbranch	15 minutes ago		6	3		...

2- Fusion des branches avec la commande « merge » :

```
nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git checkout main
Already on 'main'
M   hello.py
Your branch is up to date with 'origin/main'.

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git merge newbranch
Already up to date.

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$ git branch -d newbranch
Deleted branch newbranch (was 21c0607).

nathan@SAMSUNG-BOOK-2-PRO-NATHAN MINGW64 ~/tpgit (main)
$
```

3- Création des commits sur les 2 branches :

```
A---B---C---D---H---I ← main
      \
      E---F---G ← newbranch

A---B---C---D---H---I---J ← main
      \           /
      E---F---G ← newbranch
```

a) Commit sur la branche « main » : A, B, C, D

Commits on Jan 8, 2025

D Nathan committed 17 hours ago	3f40279	📄	<>
C Nathan committed 17 hours ago	c679b67	📄	<>
B Nathan committed 17 hours ago	52c9d0c	📄	<>
A Nathan committed 17 hours ago	80e1aa3	📄	<>

b) Commit sur la branche « newbranch » : E, F, G

newbranch			All users	All time
Commits on Jan 9, 2025				
G	Nathanser committed 20 minutes ago	14f5e1d	📄	<>
F	Nathanser committed 22 minutes ago	efa720b	📄	<>
E	Nathanser committed 23 minutes ago	3c1a778	📄	<>

c) Commit sur la branche « main » : H, I

main			All users	All time
Commits on Jan 9, 2025				
I	Nathanser committed 15 minutes ago	61305af	📄	<>
H	Nathanser committed 15 minutes ago	e30c527	📄	<>

On fait la commande git rebase ensuite