Git hub cours 1

Les SCM / VCS / RCS = logiciel qui permettent de suivre les modifications sur un projet

GitHub = outils permet de partager un dossier avec des fichiers qui contienne différent langage, mettre à jour des applis et de travailler à plusieurs dessus

SCM sert à tout sur le dev + stock le code, les diff version, qui, quand, créer des workflow, dev simultané,

Logiciel : git ; azure devops server ; subversion ; mercurial ; cvs ; aws codecommit

Peer to peer : lien, envoyer, information décentralisée, il faut internet, système protégé, copié collé chez les utilisateurs, comme la bloque chaine,

Git hub : juste une passerelle qui ne change rien à la façon de programmé, wamp / vscode / phpMyAdmin

Commande github.com :

1° New -> repository name (……) -> ø Public -> create repository

2°
echo "# GithubCours1" >> [README.md] => écrit #.... dans README.md (ça na marche pas)
git init => initialisé un dépôt local
git status => l'état du dépôt local
git add [README.md] => créer le fichier [README.md]
git commit -m "first commit" => enregistre les modifications
git branch -M main => change le nom sur la branche par main
git remote add [origin] https://github.com/ptipiouboune/GithubCours1.git => ajoute un alias
git push -u origin main (synchronise la branche main sur le dépôt origine)

Note :

Surveillé le stockage du dossier

svn enregistre des différences entre les modifications et le document de base

git à chaque version il enregistre tout « snapshot »

chekout = récupère le projet

working directory =

stage fixes =

commit = envois la modification sur GitHub avec un commentaire

add = ajoute

origin = alias

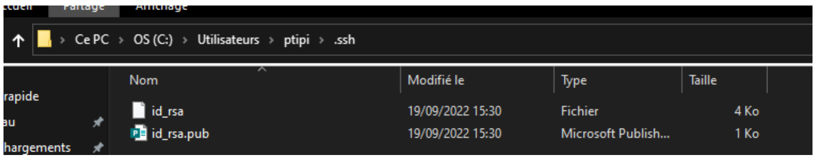
remote = adresse distante

push = envois

-u = --set upstin

Soit https soit ssh pour le lien

Travaille : générer une clef rsa =>



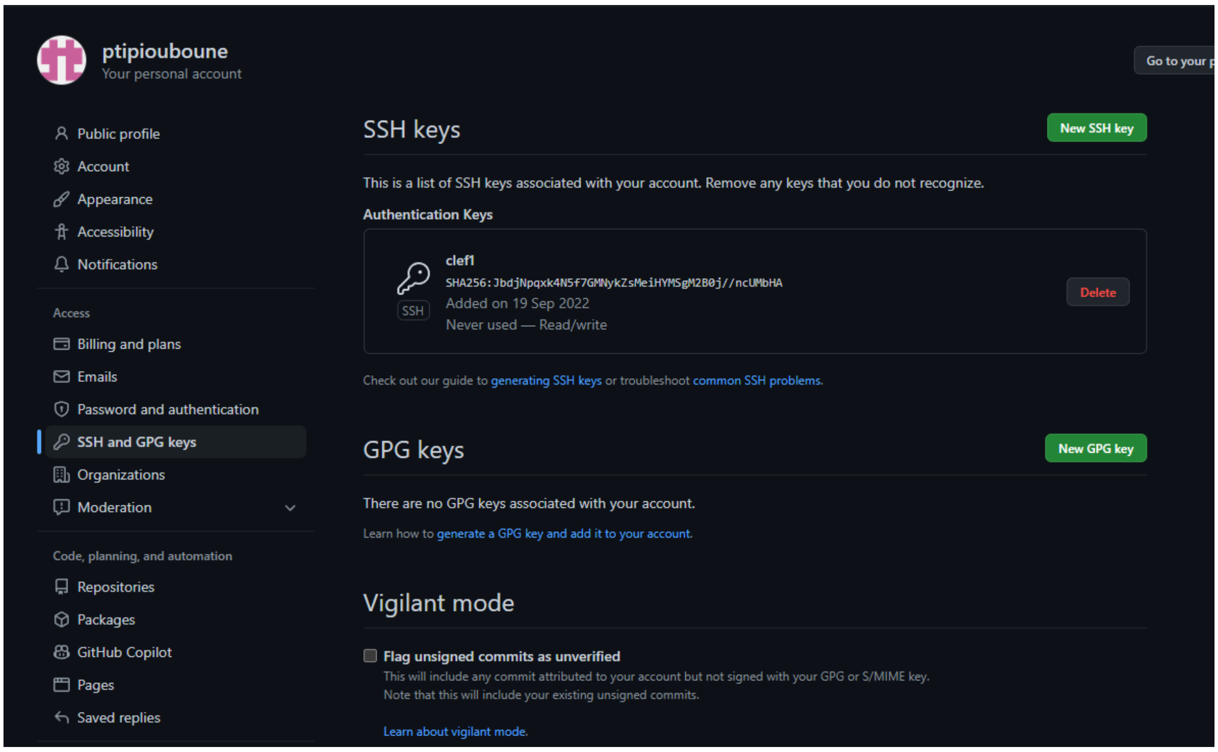
Nom	Modifié le	Type	Taille
id_rsa	19/09/2022 15:30	Fichier	4 Ko
id_rsa.pub	19/09/2022 15:30	Microsoft Publish...	1 Ko

Git cours 2

Site : Stackoverflow

Plateforme : malt

Def : git = PowerShell / github = site d'ébergement



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys

clef1
SHA256:JbdjNpqxk4N5f7GMNykZsMeiHYMSgM2B0j//ncUMbHA
SSH
Added on 19 Sep 2022
Never used — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Vigilant mode

☐ Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.

Learn about vigilant mode.

LF et CRLF sont des caractères

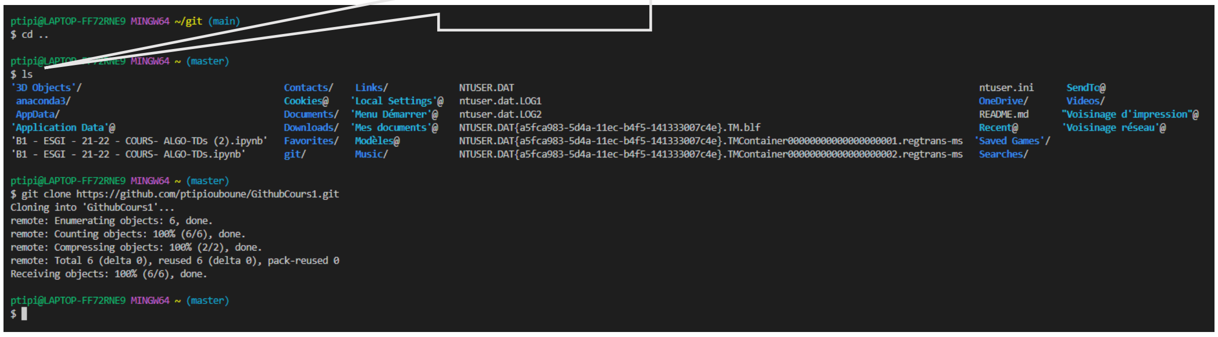
Origin = alias

Changer de clef ssh =

Sortir du dossier = cd ..

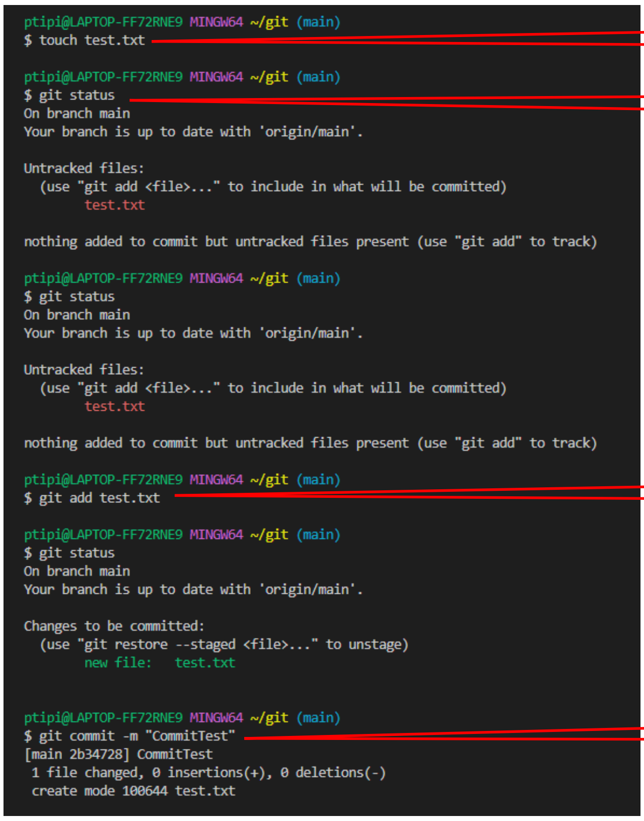
Indexé = ajouté au suivie

Clone = copie et colle le projet de GitHub à notre pc



Ls = liste

```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ cd ..

ptipi@LAPTOP-FF72RNE9 MINGW64 ~ (master)
$ ls
'3D Objects'/                                 Contacts/    Links/              NTUSER.DAT                                                                         ntuser.ini     SendTo@
 anaconda3/                                   Cookies@    'Local Settings'@    ntuser.dat.LOG1                                                                    OneDrive/      Videos/
 AppData/                                     Documents/  'Menu Démarrer'@     ntuser.dat.LOG2                                                                    README.md     "Voisinage d'impression"@
'Application Data'@                           Downloads/  'Mes documents'@     NTUSER.DAT{a5fca983-5d4a-11ec-b4f5-141333007c4e}.TM.blf                            Recent@       'Voisinage réseau'@
'B1 - ESGI - 21-22 - COURS- ALGO-TDs (2).ipynb'   Favorites/   Modèles@            NTUSER.DAT{a5fca983-5d4a-11ec-b4f5-141333007c4e}.TMContainer00000000000000000001.regtrans-ms  'Saved Games'/
'B1 - ESGI - 21-22 - COURS- ALGO-TDs.ipynb'       git/         Music/              NTUSER.DAT{a5fca983-5d4a-11ec-b4f5-141333007c4e}.TMContainer00000000000000000002.regtrans-ms   Searches/

ptipi@LAPTOP-FF72RNE9 MINGW64 ~ (master)
$ git clone https://github.com/ptipiouboune/GithubCours1.git
Cloning into 'GithubCours1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

ptipi@LAPTOP-FF72RNE9 MINGW64 ~ (master)
$ █
```



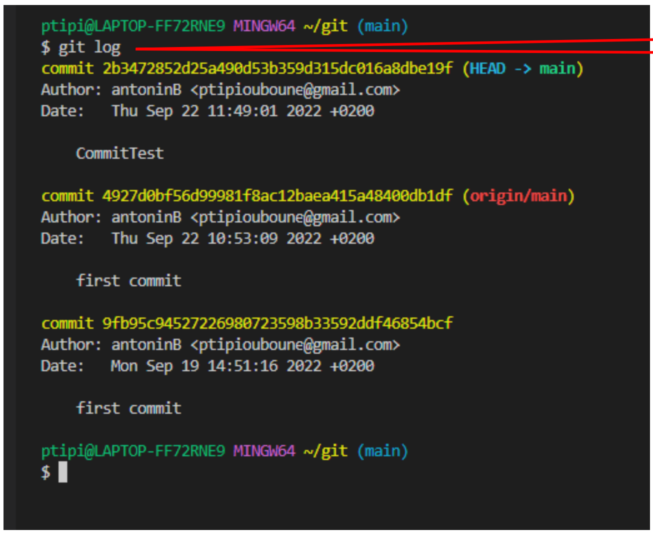
Créer un fichier

Vérifie l’état

Ajoute les modifications

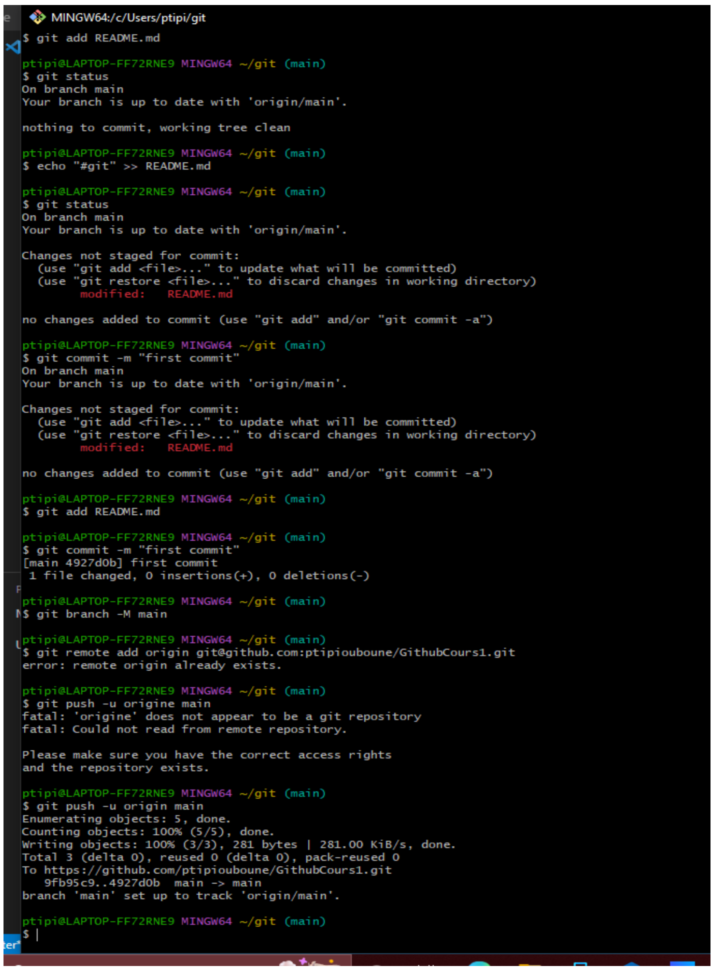
Envoie les modifications

```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ touch test.txt

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.txt

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

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.txt

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

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git add test.txt

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (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)
        new file:   test.txt


ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git commit -m "CommitTest"
[main 2b34728] CommitTest
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
```



Liste ce qui à été fait

```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git log
commit 2b3472852d25a490d53b359d315dc016a8dbe19f (HEAD -> main)
Author: antoninB <ptipiouboune@gmail.com>
Date:   Thu Sep 22 11:49:01 2022 +0200

    CommitTest

commit 4927d0bf56d99981f8ac12baea415a48400db1df (origin/main)
Author: antoninB <ptipiouboune@gmail.com>
Date:   Thu Sep 22 10:53:09 2022 +0200

    first commit

commit 9fb95c945272269807235 98b33592ddf46854bcf
Author: antoninB <ptipiouboune@gmail.com>
Date:   Mon Sep 19 14:51:16 2022 +0200

    first commit

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ █
```



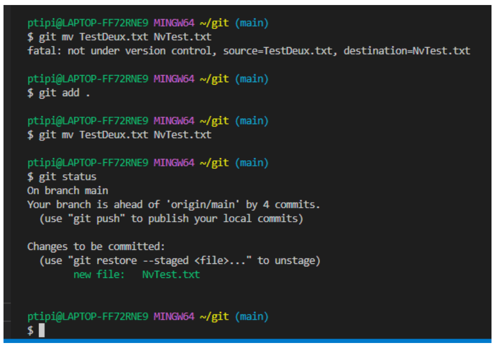
```
MINGW64:/c/Users/ptipi/git
$ git add README.md

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ echo "#git" >> README.md

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ 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

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

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git commit -m "first commit"
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

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

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git add README.md

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git commit -m "first commit"
[main 4927d0b] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git branch -M main

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git remote add origin git@github.com:ptipiouboune/GithubCours1.git
error: remote origin already exists.

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git push -u origine main
fatal: 'origine' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ptipiouboune/GithubCours1.git
   9fb95c9..4927d0b  main -> main
branch 'main' set up to track 'origin/main'.

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ |
```

Le code source est toujours le même, créer un fichier à la base permet de ne pas avoir de suivi git

Retirer un fichier de l'index = git restore --staged <nom fichier> /ou\ git reset HEAD <nom fichier> (le commit le plus récent)

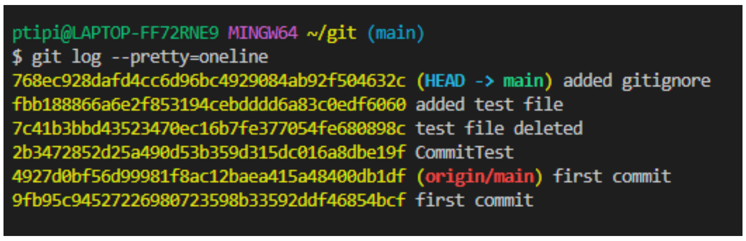
Git log => liste tout les commit

Git mv <ancien nom> <nv nom> = déplace et renomme le fichier



```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git mv TestDeux.txt NvTest.txt
fatal: not under version control, source=TestDeux.txt, destination=NvTest.txt

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git add .

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git mv TestDeux.txt NvTest.txt

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   NvTest.txt


ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ ▌
```

--prety = -p



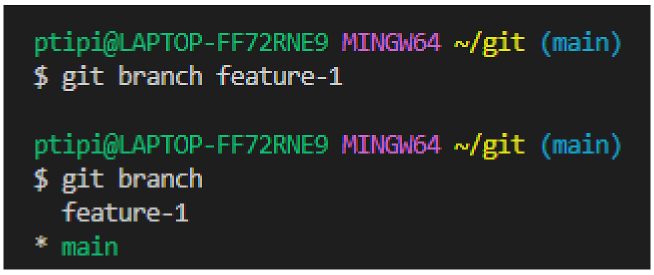
```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git log --pretty=oneline
768ec928dafd4cc6d96bc4929084ab92f504632c (HEAD -> main) added gitignore
fbb188866a6e2f853194cebdddd6a83c0edf6060 added test file
7c41b3bbd43523470ec16b7fe377054fe680898c test file deleted
2b3472852d25a490d53b359d315dc016a8dbe19f CommitTest
4927d0bf56d99981f8ac12baea415a48400db1df (origin/main) first commit
9fb95c94527226980723598b33592ddf46854bcf first commit
```

```
git log --graph --abbrev-commit --decorate --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset) %C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' –all
```

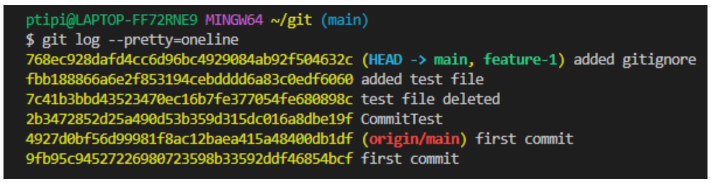
pour créer une nvl branche on va faire une copie du projet et ça n’impacteras pas le projet(commit de base)

Branche => pointeur vers un commit (copie du projet)

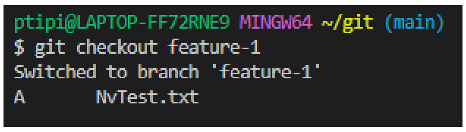
Nouvelle branche :



```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git branch feature-1

ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git branch
  feature-1
* main
```



```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git log --pretty=oneline
768ec928dafd4cc6d96bc4929084ab92f504632c (HEAD -> main, feature-1) added gitignore
fbb188866a6e2f853194cebdddd6a83c0edf6060 added test file
7c41b3bbd43523470ec16b7fe377054fe680898c test file deleted
2b3472852d25a490d53b359d315dc016a8dbe19f CommitTest
4927d0bf56d99981f8ac12baea415a48400db1df (origin/main) first commit
9fb95c945272269807235598b33592ddf46854bcf first commit
```



```
ptipi@LAPTOP-FF72RNE9 MINGW64 ~/git (main)
$ git checkout feature-1
Switched to branch 'feature-1'
A       NvTest.txt
```

(git checkout -b <nom branche>)

Git merge = fusion de deux branches

Git diff = permet de voir les modification

Git add => le fichier est indexé ; git commit => le fichier est validé

Commit / pull / push