Aubrée Romain **Beauvallet**
02/03/2026 **Semestre 2**
SLAM

Groupe 1

Atelier Git & GitHub

Contexte

Ce TP va nous expliquer comment utiliser les commandes Git et GitHub pour nos futurs projets

Exercice 1

Vérification de l’installation de **Git**

```
C:\Users\admin>git --version
git version 2.53.0.windows.1
```

Configuration de l'identité

```
C:\Users\admin>git config --global user.name "RA"

C:\Users\admin>git config --global user.email "aubree.rom@gmail.com"
```

Vérification de la configuration

```
C:\Users\admin>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.editor="C:\\Program Files\\Notepad++\\notepad++.exe" -multiInst -notabbar -nosession -noPlugin
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=RA
user.email=aubree.rom@gmail.com
```

Exercice 2

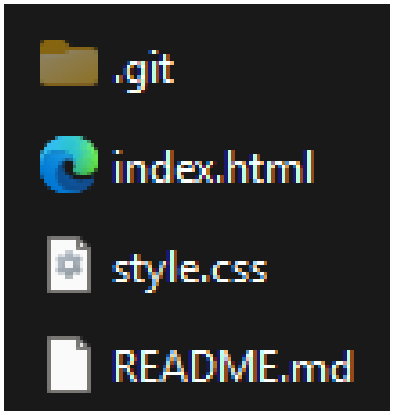
Création du dossier projet

```
C:\Users\admin>mkdir tp-git

C:\Users\admin>cd tp-git

C:\Users\admin\tp-git>
```

Initialisation de **Git**

```
C:\Users\admin\tp-git>git init
Initialized empty Git repository in C:/Users/admin/tp-git/.git/
```

Création des fichiers à mettre



Vérification de l'état (il détecte les fichiers)

```
C:\Users\admin\tp-git>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        index.html
        style.css

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

Ajout des fichiers

```
C:\Users\admin\tp-git>git add .

C:\Users\admin\tp-git>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   index.html
        new file:   style.css
```

Commit des fichiers

```
C:\Users\admin\tp-git>git commit -m "Initialisation du projet RA"
[master (root-commit) 8c7eff4] Initialisation du projet RA
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
 create mode 100644 index.html
 create mode 100644 style.css
```

Affichage de l'historique

```
C:\Users\admin\tp-git>git log --oneline
8c7eff4 (HEAD -> master) Initialisation du projet RA
```

Exercice 3

Contrôle des différences

```
C:\Users\admin\tp-git>git add "Test - Copie.txt"

C:\Users\admin\tp-git>git diff --cached
diff --git a/Test - Copie.txt b/Test - Copie.txt
new file mode 100644
index 0000000..e69de29
```

Nouveau commit

```
C:\Users\admin\tp-git>git add .

C:\Users\admin\tp-git>git commit -m "Ajout contenu page accueil"
[master c60077b] Ajout contenu page accueil
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Test - Copie.txt
```

J'ai enregistré le ***.css*** (add et commit) avec ce texte

```
Test
```

Puis je l'ai modifié et enregistré

```
Test Test x2
```

Puis j'ai exécuté la commande

```
C:\Users\admin\tp-git>git checkout -- "Test - Copie.txt"
```

Et le fichier a été rollback

```
Test
```

Exercice 4

Affichage des branches

```
C:\Users\admin\tp-git>git branch
* master
```

Création d'une nouvelle branche (elle switch automatiquement dessus)

```
C:\Users\admin\tp-git>git checkout -b navbar
Switched to a new branch 'navbar'
```

Fusion de la branche “navbar” avec “master”

```
C:\Users\admin\tp-git>git checkout master
Switched to branch 'master'

C:\Users\admin\tp-git>git merge navbar
Already up to date.
```

Exercice 5

Tentative de merge des branches où ils modifie le même fichier mais avec des informations différentes

```
C:\Users\admin\tp-git>git merge navbar
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

Git a généré ça

```
<<<<<<< HEAD
test
=======
tefzfggfegsdq
>>>>>>> navbar
```

J’ai modifier le fichier pour régler le conflits puis je l’ai commit

```
C:\Users\admin\tp-git>git add .

C:\Users\admin\tp-git>git commit -m "Resolution conflit"
[master 1c795d9] Resolution conflit
```

Exercice 6

Liaison du fichiers **Git** a un projet **GitHub**

```
C:\Users\admin\tp-git>git remote add origin https://github.com/RomAub/AtelierGithub.git

C:\Users\admin\tp-git>git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 4 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (30/30), 2.30 KiB | 336.00 KiB/s, done.
Total 30 (delta 11), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (11/11), done.
To https://github.com/RomAub/AtelierGithub.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Récupération de ce qu'il y a sur **GitHub** pour le mettre sur le PC

```
C:\Users\admin\tp-git>git pull
Already up to date.
```

Clonage du projet et de l'historique

```
C:\Users\admin\tp-git>git clone https://github.com/RomAub/AtelierGithub
Cloning into 'AtelierGithub'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 30 (delta 11), reused 30 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), done.
Resolving deltas: 100% (11/11), done.
```

Exercice 7

Pourquoi Git est indispensable ?

C'est un "bouton de sauvegarde" géant. On peut revenir à n'importe quelle version précédente si tu casses tout. Plusieurs personnes travaillent sur le même code sans s'écraser leurs fichiers.

Quand créer une branche ?

Lorsqu'on veut travailler sur quelque chose de nouveau, on ne travaille jamais sur la branche principale (main/master).

Différence pull / push ?

push = On envoie son travail de son PC vers **GitHub**.

pull = On récupère le travail des autres depuis **GitHub** vers son PC.

Que se passe-t-il sans commit ?

Rien n'est sauvegardé et tout reste sur l'ordinateur.

Conclusion

Ce TP nous a montré à utiliser **Git**, ce qui nous servira pour nos futurs projets en commun.