

TP1 - Maîtrise de Git

Partie 1 : Préparation de l'environnement Git

1. Création de clé SSH

```
cheml@Shaima-Laptop MINGW64 ~ (master)
$ ssh-keygen -t rsa -b 4096 -C shaima.chemli@polytechnicien.tn
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/cheml/.ssh/id_rsa):
Created directory '/c/Users/cheml/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/cheml/.ssh/id_rsa
Your public key has been saved in /c/Users/cheml/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vn0tDeF0KAofBRmKgnTbXhETZyxAkXSbdQTL0HLK7k shaima.chemli@polytechnicien.
tn
The key's randomart image is:
+---[RSA 4096]-----+
| o+=B*+= . |
|...oo@.= o o |
```

2. Configuration de Git

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git config --global user.name shaima chemli

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git config --global user.email shaimachemli@gmail.com
```

3. Connexion SSH aux dépôts distants

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ ssh -T git@github.com
Hi shaimachemli21! You've successfully authenticated, but GitHub does not provide shell access.
```

4. Comment vérifier la configuration actuelle de Git sur votre machine, notamment le nom d'utilisateur et l'adresse e-mail ?

git config --list

5. Comment modifier votre adresse e-mail si vous l'avez mal configurée lors de l'installation de Git ?

git config --global user.email nouvelle_adresse@email.com

Partie 2: Création d'un nouveau projet

1. Créer projet

The screenshot shows the GitHub 'Create a new repository' interface. It has two main sections: 'General' and 'Configuration'. In the 'General' section, the 'Owner' is 'shaimachemli21' and the 'Repository name' is 'TP1_Git1', with a green checkmark indicating it is available. There is a text input for 'Description' with a 0/350 character limit. The 'Configuration' section includes 'Choose visibility' set to 'Public', 'Add README' with a toggle switch turned 'On', 'Add .gitignore' set to 'No .gitignore', and 'Add license' set to 'No license'. A green 'Create repository' button is at the bottom right.

2. Notez l'URL SSH du projet.

The screenshot shows the GitHub 'Clone' dialog. It has tabs for 'HTTPS', 'SSH', and 'GitHub CLI'. The 'SSH' tab is selected, and the URL 'git@github.com:shaimachemli21/TP1_Git.git' is displayed in a text box. A copy icon is to the right of the text box. Below the text box, it says 'Use a password-protected SSH key.'

3. Clonez le projet en utilisant l'URL SSH

```
chem1@Shaima-Laptop MINGW64 ~ (master)
$ git clone git@github.com:shaimachemli21/TP1_Git.git
Cloning into 'TP1_Git'...
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.
```

4. Accéder dans le projet

```
chem1@Shaima-Laptop MINGW64 ~ (master)
$ cd TP1_Git
```

6. Si vous avez oublié de créer un fichier README.md lors de l'initialisation du projet, comment pouvez-vous l'ajouter après coup et committer les changements ?
 - Crée le fichier avec la commande : `echo "# Fraud Detection MLOps Project" > README.md`
 - Ajoute-le à Git
 - Fais un commit: `git commit -m "Ajout du fichier README.md"`
 - Envoie sur GitHub: `git push origin main`
7. Comment définir un dépôt distant si vous n'en avez pas configuré un lors de la création du projet ?
 - Ajouter le dépôt distant: `git remote add origin <url>`
 - Vérifier: `git remote -v`
 - Pousser la branche principale: `git push -u origin main`

Partie 3 : Concepts de base de Git

1. Travailler avec les fichiers

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ touch eda.py

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ echo "import pandas as pd\nndf = pd.read_csv('data.csv')\nprint(df.head())" > eda.py

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git add eda.py
warning: in the working copy of 'eda.py', LF will be replaced by CRLF the next time Git touches it

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git commit -m "Premier commit : ajout de eda.py"
[main 6608786] Premier commit : ajout de eda.py
1 file changed, 1 insertion(+)
create mode 100644 eda.py
```

2. Historique des commits

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git log
commit 6608786bd3ecb95c1206b3477350d8db6cc24920 (HEAD -> main)
Author: shaima chemli <shaima.chemli@polytechnicien.tn>
Date: Thu Oct 16 12:33:42 2025 +0100

    Premier commit : ajout de eda.py

commit 6c4da89b1d2c1f48b397c2b0cf82c49d76be3761 (origin/main, origin/HEAD)
Author: shaimachemli21 <151739737+shaimachemli21@users.noreply.github.com>
Date: Thu Oct 16 12:08:57 2025 +0100

    Initial commit
```

3. Comment annuler les modifications locales d'un fichier avant de les ajouter à l'index ?

git restore eda.py

4. Comment visualiser les fichiers qui sont prêts à être committés dans Git (staging) ?

git status

Partie 4 : Collaborer sur Git

1. Créer une nouvelle branche pour une fonctionnalité

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git branch experiment-eda

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git checkout experiment-eda
Switched to branch 'experiment-eda'
```

2. Effectuer des modifications et pousser vers le dépôt distant

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git add .

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git commit -m "Modification de experiment-eda"
On branch experiment-eda
nothing to commit, working tree clean

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git push origin experiment-eda
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 358 bytes | 179.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'experiment-eda' on GitHub by visiting:
remote: https://github.com/shaimachemli21/TP1_Git/pull/new/experiment-eda
remote:
To github.com:shaimachemli21/TP1_Git.git
 * [new branch] experiment-eda -> experiment-eda
```

3. Gestion des conflits

```
Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git checkout experiment-eda
Already on 'experiment-eda'

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git commit -am "Modification dans experiment-eda"
On branch experiment-eda
nothing to commit, working tree clean

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git commit -am "Modification dans master"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git merge experiment-eda
Already up to date.

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git add .

Chem1@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git commit -m "Résolution du conflit"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
```

4. Comment suivre (track) un dépôt distant et récupérer toutes les branches de ce dépôt ?

- Pour ajouter un dépôt distant si ce n'est pas déjà fait : **git remote add origin <url_du_dépôt>**
- Pour récupérer toutes les branches du dépôt distant : **git fetch --all**
- Pour voir les branches locales et distantes : **git branch -a**

5. Comment supprimer une branche locale après l'avoir fusionnée dans master ?

- git branch -d experiment-eda

Partie 5 : Rebase d'une branche sur 'master'

1. Passer sur la branche master et la mettre à jour :

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git pull origin main
From github.com:shaimachemli21/TP1_Git
* branch          main      -> FETCH_HEAD
Already up to date.
```

2. Changer de branche pour celle à intégrer :

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git checkout experiment-eda
Switched to branch 'experiment-eda'
```

3. Rebaser la branche experiment-eda sur master :

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git rebase main
Current branch experiment-eda is up to date.
```

4. Retourner sur la branche master

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (experiment-eda)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

5. Fusionner sans commit supplémentaire :

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git merge experiment-eda
Already up to date.
```

6. Pousser les modifications sur le dépôt distant :

```
cheml@Shaima-Laptop MINGW64 ~/TP1_Git (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:shaimachemli21/TP1_Git.git
6c4da89..6608786  main -> main
```

7. Comment interrompre un rebase en cours si vous avez commis une erreur ?

git rebase --abort

8. Comment lister les commits qui vont être rebasés avant de lancer un rebase ?

git log master..experiment-eda