

## TP 1: Commandes et opérations sur GitHub

L'objectif de ce TP est de suivre l'évolution d'un projet, localement et à distance. Généralement, en génie logiciel un programme est développé par un groupe de développeur. Pour que les différents membres d'une équipe communiquent facilement et pour gérer les versions modifiées par chaque membre, plusieurs outils de travail collaboratifs sont utilisés. Dans ce TP, nous allons choisir **GitHub** comme un outil de travail collaboratif.

### Les opérations de bases de Git

**Clone:** *git clone <url\_remote\_repository>* permet de copier le contenu du dépôt distant dans le dossier courant. Cette commande est utilisée lorsqu'on n'a pas encore de dépôt local du projet.

**Initialize:** *git init* permet de créer un dépôt local git vide ou réinitialiser un existant.

**Add:** *git add <file>* permet la mise à jour du contenu de l'arborescence de travail avant l'exécution de la commande commit.

**Commit:** *git commit -a -m <message>* permet d'enregistrer les modifications effectuées sur un projet dans un dépôt local.

**Push:** *git push <remote-name> <branch-name>* permet de partager votre projet dans un dépôt distant.

**Pull:** *git pull origin <branch-name>* Cette commande met à jour le dépôt local à partir du dépôt distant.

### Manipulation

Dans ce TP, vous allez installer git sur votre ordinateur, créer un dépôt git distant sur GitHub et apprendre à utiliser ses fonctionnalités.

Après avoir installé Git sur votre machine, l'étape suivante consiste à créer un compte GitHub gratuit.

1. Visiter la page officielle de création de compte : Rejoindre GitHub
2. Choisir un nom d'utilisateur, saisir votre adresse électronique et choisir un mot de passe.
3. Accepter ou refuser de recevoir des mises à jour et des annonces en cochant/décochant la case Préférences de messagerie.
4. Vérification que ce n'est pas un robot.
5. Cliquer sur Créer un compte.

6. GitHub envoie un code de lancement à l'adresse électronique spécifiée. Passer le code pour vérification.

## **Partie I: Utilisation de l'éditeur «Visual Studio Code»**

### **Étape 1: Initialisation du projet en tant que référentiel Git**

Afin d'initialiser votre projet, vous devez suivre ces étapes:

1. Dans le panneau de gauche de VS Code, ouvrir l'onglet **Contrôle de Code Source**:
2. Cliquer sur **Ouvrir un dossier** L'explorateur de fichiers du répertoire actuel est ouvert.
3. Choisir votre répertoire de projets, puis cliquer sur **Ouvrir**.

4. Choisir **Initialiser un dépôt**

\*\*\*Votre système de fichiers, contient maintenant un répertoire.git.

\*\*\*Utiliser le terminal afin d'explorer le répertoire du projet.

5. Créer le fichier «index.html».

Dans le panneau **Contrôle de Code Source**, ce fichier apparaît avec la lettre **U**(*untracked file*).

➔ En cliquant sur l'icône **plus (+)**, la lettre à côté du fichier se transforme en **A**.

Donc, ce fichier qui a été bien ajouté au référentiel.

6. Pour la validation des modifications, il faut taper un message de commit en cliquant sur l'icône **check** dans la zone de saisie située en haut du panneau **Contrôle de Code Source**.

### **Étape 2: Modification du projet**

1. Ajouter du contenu au fichier index.html,
2. Cliquer sur l'icône **plus (+)** pour la modification du fichier.  
Une lettre «**M**»apparaîtra qui représente la modification du dossier.
3. Afin de valider ces changements, cliquer sur l'icône **check** (pour créer un commit).

\*\*\*Un bouton pour synchroniser les modifications apparaîtra.

\*\*\*En cliquant sur ce bouton, ces modifications seront enregistrées.

4. Une autre façon pour enregistrer ces modifications dans le dossier distant consiste à choisir la commande «Push» se trouvant dans le panneau **Contrôle de Code Source**

### Étape 3 - Création des branches

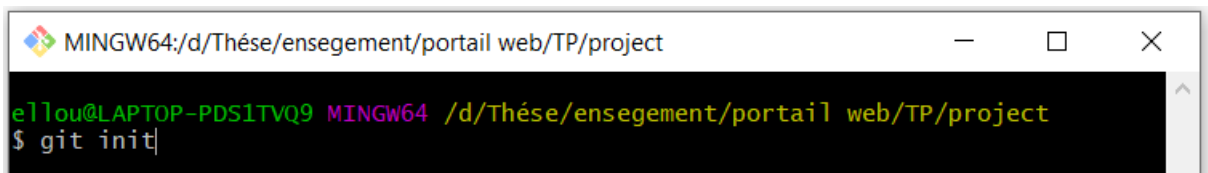
1. Afin de créer et de changer de branche, cliquer sur l'icône «contrôle de source»:
2. Afin de créer la branche, entrer son nom.
3. Créer la nouvelle branche «test».
4. Modifier le fichier«index.html».
5. Enregistrer les modifications et revenir à la branche master.

### Méthode 2: Utilisation des commandes Git:

#### Étape1: Créer un dépôt Git local

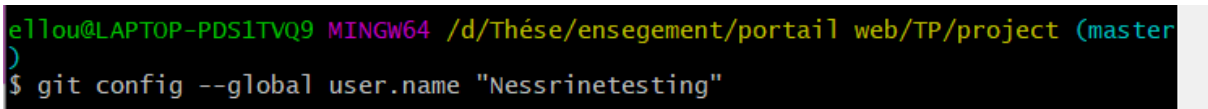
Après avoir installé ou mis à jour Git, l'étape suivante consiste à créer un dépôt Git local.

1. Ouvrir un terminal Git Bash et aller dans le répertoire où vous souhaitez conserver le projet sur votre machine locale. Par exemple :
2. Créer un dépôt Git dans le dossier sélectionné en exécutant la commande git init :

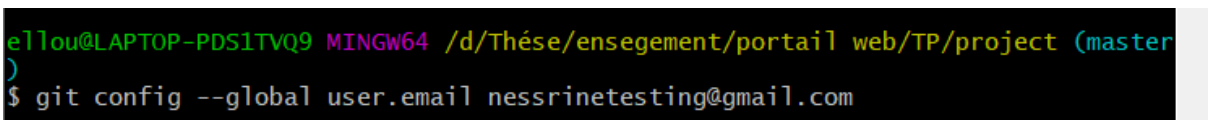


```
MINGW64:/d/Thèse/ensegement/portail web/TP/project
e1lou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/ensegement/portail web/TP/project
$ git init
```

3. Configurer votre compte Github avec l'espace de travail local sur votre ordinateur.



```
e1lou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/ensegement/portail web/TP/project (master)
$ git config --global user.name "Nessrinetesting"
```



```
e1lou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/ensegement/portail web/TP/project (master)
$ git config --global user.email nessrinetesting@gmail.com
```

Vous avez maintenant configuré avec succès un dépôt Git local.

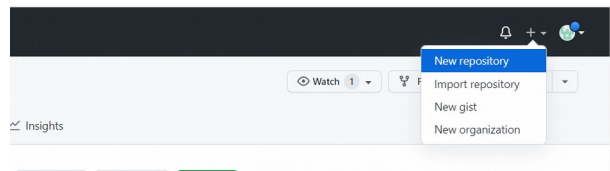
#### Étape 2: créer un nouveau dépôt sur GitHub

GitHub permet la sauvegarde d'une trace du code lorsque vous travailler avec une équipe et vous modifier le code du projet de manière collaborative.

Pour créer un nouveau dépôt sur GitHub :

1. Connecter à la page d'accueil de GitHub sur <https://github.com/>.

2. Cliquer sur l'option Nouveau dépôt sous le signe +, puis cliquer sur New repository.



3. Nommer votre référentiel (repository) appelée TP1, saisir une description significative et choisir un paramètre de confidentialité.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [cautious-potato?](#)

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set **main** as the default branch. Change the default name in your [settings](#).

[Create repository](#)

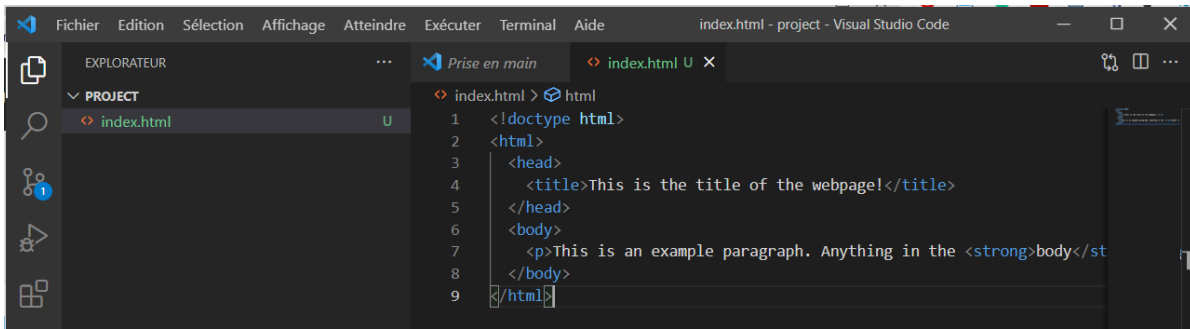
4. Cliquer sur le bouton «Create repository».

GitHub vous permet d'ajouter un dépôt existant que vous avez créé localement. Pour héberger un dépôt local de votre machine vers GitHub à travers la commande «git remote » :

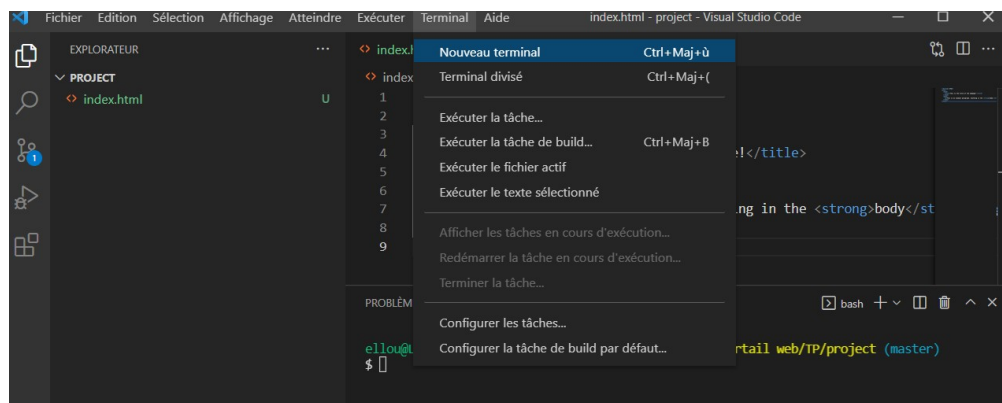
```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/enseignement/portail web/TP/project (master)
$ git remote add origin https://github.com/Nessrinetesting/TP1.git
```

### Étape 3: Ajouter un fichier dans un projet

1. Ouvrir l'éditeur VS Code et créer un nouveau fichier appelé index.html.



## 2. Ouvrir un nouveau terminal « Git Bash » sous VS Code



## 3. Vérifier les fichiers suivis par Git en exécutant la commande « git status » :

```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/enseignement/portail web/TP/project (master)
$ git status
On branch master

No commits yet

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

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

Ajouter le fichier index.html en utilisant la commande « git add ».

## Étape 4: Créer un commit

Après avoir ajouté les fichiers spécifiés à l'environnement de transit, référencier les fichiers modifier en utilisant la commande « git commit ».

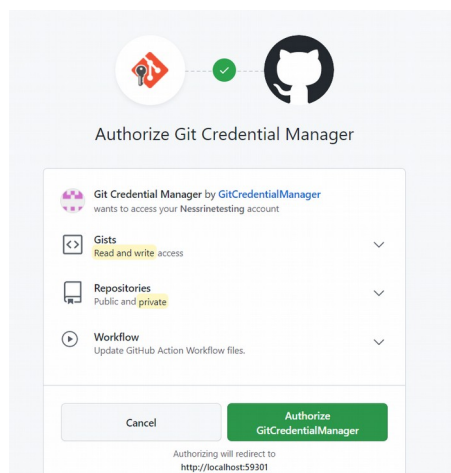
```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/enseignement/portail web/TP/project (master)
$ git commit -m "version 1"
[master (root-commit) b9e7b59] version 1
1 file changed, 9 insertions(+)
create mode 100644 index.html
```

## Étape 5: Ajouter les paramètres d'accès au GitHub sous VS Code

1. Utiliser la commande **git config --global credential.helper store** pour enregistrer les paramètres de connexion sous VS Code.

```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/enseignement/portail web/TP/project (
master)
$ git config --global credential.helper store
```

2. Cliquer sur Autoriser.



3. Synchroniser le dossier local TP1 (origin) avec le dossier distant TP1 (remote) dans la branche master en utilisant la commande «**git push origin master**».

```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/enseignement/portail web/TP/project (
master)
$ git push origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 382 bytes | 382.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Nessrinetesting/TP1/pull/new/master
remote:
To https://github.com/Nessrinetesting/TP1.git
 * [new branch]      master -> master
```

## Étape 6: Ajouter une nouvelle branche «groupe1»

1. Créer une nouvelle branche appelée groupe1

```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/ensegement/portail web/TP/project (
master)
$ git branch groupe1
```

2. Ajouter la branche groupe 1 dans le dossier remote en utilisant la commande «**git push origin groupe1**».

```
ellou@LAPTOP-PDS1TVQ9 MINGW64 /d/Thèse/ensegement/portail web/TP/project (
master)
$ git push origin groupe1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'groupe1' on GitHub by visiting:
remote:   https://github.com/Nessrinetesting/TP1/pull/new/groupe1
remote:
To https://github.com/Nessrinetesting/TP1.git
 * [new branch]      groupe1 -> groupe1
```

## Étape 7: Invitation des collaborateurs sur le projet TP1

Cliquer sur le répertoire **TP1** et sélectionner l'onglet **Settings** puis **collaborators** pour ajouter des Coéditeurs sur ce projet.

