

# Utilisation de GitHub et Visual Studio Code

## OBJECTIFS

Voici les objectifs de ce cours :

- ☒ Créer un compte GitHub
- ☒ Télécharger GitBash
- ☒ Configurer la connexion en SSH
- ☒ Télécharger Visual Studio Code
- ☒ Utiliser les commandes usuelles de Git

## TD1

### Exercice 1 - GitHub Online

Toutes les questions de cette exercice sont à réaliser sur la version GitHub online.

1. Créer un compte sur GitHub si vous n'avez pas encore de compte
2. Créer un repository GitHub **public** appelé `iut_sd3_accidentss` sur le workspace online avec un fichier README.md par défaut ([plus d'info ici \(https://docs.github.com/fr/repositories/creating-and-managing-repositories/creating-a-new-repository#creating-a-new-repository-from-the-web-ui\)](https://docs.github.com/fr/repositories/creating-and-managing-repositories/creating-a-new-repository#creating-a-new-repository-from-the-web-ui)).

Pour personnaliser le fichier README.md, vous pouvez vous aider de [ce lien \(https://www.markdownguide.org/cheat-sheet/\)](https://www.markdownguide.org/cheat-sheet/).

3. Modifier le fichier README.md en ajoutant un titre de niveau 2

```
## Titre 2
```

4. La modification sauvegardée apparaît sous la forme d'un `commit`
5. Créer une branche appelée `votre_prénom` depuis la branche principale.
6. Dans la branche `votre_prénom`, ajouter un fichier `test.py` avec ce bout de code :

```
## print("hello")
```

7. Créer une branche appelée *votre\_prénom\_DEV* depuis la branche principale.

8. Dans la branche *votre\_prénom\_DEV* ajouter un fichier *test.py* avec ce bout de code :

```
## print("Hola")
```

Nous avons donc 3 branches :

- ☒ la branche principale *main*
- ☒ la branche *votre\_prénom*
- ☒ la branche *votre\_prénom\_DEV*

9. Créer une *pull request* de la branche *votre\_prénom* sur la branche principale *main*

10. Merger la *pull request* pour fusionner les branches sans supprimer la branche *votre\_prénom*. Aucun conflit n'apparaît entre les deux branches.

11. Créer une *pull request* de la branche *votre\_prénom\_DEV* sur la branche principale *main*

12. Merger la *pull request* pour fusionner les branches sans supprimer la branche *votre\_prénom\_DEV*. Un conflit apparaît entre les deux branches. Il faut le résoudre en choisissant une des deux syntaxes du fichier *test.py* qui a été créée dans les deux branches *votre\_prénom* et *votre\_prénom\_DEV* puis en cliquant sur *mark as resolved*.

## Exercice 2 - Configurer GitHub en local

Dans cet exercice, nous allons utiliser des commandes Git avec GitBash.

1. Télécharger Git et GitBash

2. Ouvrir GitBash et télécharger votre repository public via son adresse HTTPS avec la commande :

```
git clone https://github.com/asardell/IUT_SD3.git
```

Le dossier est téléchargé dans le répertoire courant. Cependant, nous ne pouvons pas faire de modification en local car nous ne sommes pas connectés en tant qu'admin. Nous allons le faire via une connexion SSH. 3. Créer un dossier *IUT\_SD3\_Github* sur votre bureau 4. Ouvrir GitBash et placer vous sur le dossier créé avec :

```
cd Desktop/IUT_SD3_Github
```

5. Générer la clé SSH avec :

```
ssh-keygen
```

6. Saisir *id\_rsa* comme filename pour sauvegarder la clé ssh

7. Entrer une passphrase qui permettra de se connecter à votre compte GitHub à chaque connexion
8. Ajouter la clé en tant que nouvelle clé dans les paramètres de votre compte GitHub online ([voir lien utile \(https://linuxhint.com/clone-repo-with-ssh-key-in-git/\)](https://linuxhint.com/clone-repo-with-ssh-key-in-git/)). Vous avez le choix entre la clé public ou privée selon la visibilité de votre compte.
9. Tester la connexion SSH avec :

```
eval "$(ssh-agent -s)"
```

10. Puis ajouter le fichier `id_rsa` à la clé SSH de votre session Git

```
ssh-add ./id_rsa
```

11. Pour vérifier si la connexion est fonctionnelle, changer de chemin vers le dossier dans lequel vous souhaitez cloner votre repository :

```
cd Desktop/IUT/SD3/Anthony
```

12. Utiliser l'adresse SSH de votre repository disponible sur votre compte online

```
git@github.com:asardell/IUT_SD3.git
```

## Exercice 3 - Tester les commandes usuelles

Dans cet exercice, nous allons tester plusieurs commandes Git, pensez à vérifier la bonne exécution de vos commandes sur votre compte online.

1. Se placer sur votre repository local avec :

```
cd Desktop/IUT/SD3/Anthony/IUT_SD3
```

2. Lister le nombre de branch avec :

```
git branch
```

3. Créer une branche `DEV`.

```
git branch DEV
```

4. Ajouter dans cette branche les 4 fichiers `.csv` du projet dans un dossier appelé `data`
5. Sauvegarder les modifications effectuées pour le prochain `commit` avec :

```
git add .
```

6. Enregistrer votre premier `commit` en local avec :

```
git commit -m "[message descriptif]"
```

7. Envoyer tous les commits de la branche locale vers GitHub online avec :

```
git push origin DEV
```

Pour éviter les erreurs de manipulation, on préfère effectuer les opérations `pull requests` et `merge` uniquement sur GitHub online.

8. Créer une `pull request` de la branche `DEV` sur la branche principale `main`

9. Merger la `pull request` pour fusionner les branches sans supprimer la branche `DEV`.

## Exercice 4 - Exploration de données avec Python

Dans cet exercice, nous allons nous familiariser avec les données du projet en Python mais en travaillant sur des scripts `.py` et pas des notebooks.

1. Installer Python
2. Installer Visual Studio Code
3. Créer un script python appelée `step1.py` dans la branche `DEV`
4. Dans ce script, lire les 4 fichiers et les fusionner dans un seul avec des commandes de la librairie `pandas`.
5. Si des librairies nécessaires ne sont pas installées, exécuter la commande ci-dessous dans un terminal dans Visual Studio Code ou en dehors :

```
pip install pandas
```

Pour vérifier qu'une librairie est installée, exécuter la commande :

```
pip list
```

6. Exporter le dataframe obtenu dans `/step1/merged_data.csv`
7. Enregistrer ces modifications vers le repository distant.

## LIENS UTILES

Voici quelques liens utiles qui pourrait vous aider durant ce projet :

- Markdown Cheat Sheet (<https://www.markdownguide.org/cheat-sheet/>)
- GitHub Cheat Sheet (<https://training.github.com/downloads/fr/github-git-cheat-sheet.pdf>)
- GitHub Get Started (<https://docs.github.com/fr/get-started/quickstart/hello-world>)
- GitHub Set up (<https://docs.github.com/fr/get-started/quickstart/set-up-git>)
- Bash Cheat Sheet (<https://github.com/RehanSaeed/Bash-Cheat-Sheet>)
- Tutoriel GitHub SSH (<https://linuxhint.com/clone-repo-with-ssh-key-in-git/>)