Workshop git/github

1. Créez un nouveau répertoire sur votre compte GitHUb et nommez le test\_git.

* **Go to github and log in to your account , Click the new repository  button in the top-right and named the directory “ test\_git after that click the “Create repository” button.**

1. Copiez l'URL de ce répertoire sur votre machine locale on utilisant l'option SSH.

* **enter the Dashboard and Select a repository from the Your Repositories list after that Click the Clone or download button and copy the repository link (for SSH).**

1. En ligne de commande sur votre machine locale, naviguez jusqu'à l'endroit où vous souhaitez stocker ce projet, puis clonez votre répertoire sur GitHub sur votre ordinateur avec git clone suivi de l'URL que vous avez copiée à la dernière étape. La commande complète devrait ressembler à git clone[git@github](<mailto:git@github.com).com:nom-utilidsateur/test_git.git>.

* **First step write command git clone after that past the URL copied**

1. Déplacez vous sur le répertoire cloné et vérifiez si la connexion s'est bien établie avec le répertoire sur GitHub: git remote -v

* **We use git remote-v to check that command has worked**

1. Décrire le résultat de cette commande?

* **Origin (push)+repository URL**
* **Origin (fetch)+repository URL**

1. Créez un nouveau fichier dans le dossier git\_test appelé README.md.

* **touch git\_test/README.md**

1. Tapez la commande  dans votre terminal et décrivez le résultat de la commande. **git status**

* **After type this command we noticed it using to shows the current state of your Git working directory in this situation the interpreter tell us there’s untracked file**

1. Tapez la commande git add README.md.

$ git add README.md and press “Enter"

1. Tapez la commande git status. Que remarquez vous? et pourquoi?

* **After use “git add “command and use “git status” command we noticed that the interpreter tell us we staged the file and refer to us if we want to unstage to use command git rest head**

1. Tapez la commande git commit -m "Ajouter README.md".

* **$ git commit -"Ajouter README.ma"**

1. Tapez la commande git status.

* **$ git status**
* **This time interpreter tell us that we have to use git push to publish our local commits**

1. Avec vos propre mots, décrire le workflow que vous venez de pratiquer de l'étape 6 à 11.

* **After your code has been tested and is complete, add your changes to the staging area. This means that your files are tracked by Git. So that run “git status” command to confirm that your files have been added to the staging area after that we Commit your changes back to the repository by <<git commit -m"sort message about the change” >>**

1. Tapez git log et décrivez le résultat de la commande, ainsi le fonctionnement de la commande git log.

* **“$ git log “ this command displayed all commits in repository’s history**

1. Créez un nouveau fichier hello\_world.txt dans le répertoire test\_git.

* **$ touch test\_git/hello\_world\_tx**t

1. Vérifiez le statut de Git.

* **$ git status**

1. Ouvrir le fichier README.md avec votre éditeur de texte et ajouter une petite présentation de vous puis enregistrer les changement et fermer le fichier.

* **Go to README.md file and a open it and write “hiii every one"**

1. Vérifiez le statut de Git.

* **After checking the state git we noticed on it note tell us that file is modified**

1. Ajoutez README.md à la staging area avec git add ..

* **$ git add .**

1. Pouvez-vous deviner quel sera le statut de Git maintenant? Quelle est la différence entre git add . et git add README.md.

* **After type git status we’ll notice the interpreter tell us that file staged**
* **There’s no different between git add . And git add README .md because “git add .”stages new files and modifications, without deletions**

1. Valider tous les fichiers qui se trouvent dans la staging area et ajoutez un message de validation descriptif git commit -m "votre message".

* **$ git commit -m “modified README.md”**

1. Vérifiez le status de git maintenant.
2. Tapez git push origin main

* **$ git push origin main**

1. Verifiez le statut de git
2. Jetez un coup d'oeil sur votre répertoire GihHub. Des remarques?