**CONFIGURATION DE GIT**

$ git config --global user.name "cheikh" : pour configurer mon nom

$ git config --global user.email "cheikhjoop2002@gmail.com» : configurer mon email

$ git config --global color.ui auto

$ git config -l : pour voir la configuration

**COMMANDES STAGING AREA**

$ git init . : Initialiser un repository git

$ git status : nous montre les modifications que nous avons apporté

$ git add index.html index.js style.css : pour ajouter nos trois fichiers en zone de préparation

$ git add . : ajouter tous les fichiers qui se trouve dans le dossier courant

$ git add --all ou $ git add -A : Ajouter les élément qui sont dans le dossier

courant et les dossiers parent

$ git rm --cached index.html index.js style.css : pour les faire ressortir

$ git rm -r --cached . : pour ressortir tous les fichiers

$ git reset : to remove all files from the staging area

**CAMMANDES GIT COMMIT**

A git commit is a save point

$ git commit -m "my first commit" : pour créer un commit

---- Pour regarder notre commit

1. $ git log

On copie la hache et on fait :

1. $ git show on colle la hache

$ git diff : compare les modifications

$ vi index.js : pour éditer le fichier index.js

$git reset --soft HEAD~1 : Pour supprimer le commit num 1

$ rm -rf .git : Reinitialiser tous les git commit

**GITHUB**

GITHUB is a remote server

GitHub est une plateforme d’hébergement et de collaboration sur des référentiels git.

Un dépôt git est un dépôt héberger par Github

$ git push -u origin main : Ajouter le depot dans le GITHUB

**Ajouter un nouveau fichier sur un github existant**

Je veux ajouter un fichier Angular.txt sur sur mon repository learning\_git

Etape :

Vi Angular.txt : Pour créer le fichier sur main

Git add . : Pour l’ajouter dans notre Area

Git commit -m « nom de mon nouveau commit »

Git push : seulement pas besoin de git push -u origin main car on est déjà sur main(déjà créer)

SI on modifie un fichier sur gitthub, On tape la commande

Git pull : pour enregistrer la modification

**BRANCH**

En Git, une branche (Branch en anglais) est une ligne de développement indépendante qui représente une séquence particulière de modifications apportées à un ensemble de fichiers.

$ git branch -r : voir la liste des branches

$ git branch -a

$ git branch my\_new\_branch : Créer une nouvelle branche appelée « my\_new\_branch »

$ git checkout my\_new\_branch : Pour changer de branche

$ git push -u origin my\_new\_branch : Ajouter la branche sur Github

$ git checkout -b toDelite : Créer directement une branche et se déplacer la dessus

$ git branch -d toDelite : Pour supprimer cette branche

On va sur le menu PULL REQUESTS pour merge les DEUX branches et supprimer la branche de modification. Comme ça, toutes les modifications seront insérer sur la principale.

$ git pull

En suite

$ git log –online : Pour voir les modifications du merge