

Exercice : Gestion d'un projet avec Bash, Git et GitHub

Contexte

Vous êtes un développeur qui démarre un nouveau projet et souhaitez le gérer avec **Git** et **GitHub** tout en utilisant les commandes de base en **Bash**. Votre objectif est de créer un fichier, le versionner avec Git, travailler sur une branche séparée et enfin publier votre projet sur GitHub.

Objectifs

- Manipuler des fichiers et répertoires avec **Bash**.
- Gérer un dépôt **Git** local (initialisation, commit, branche).
- Fusionner une branche et supprimer les branches inutiles.
- Publier le projet sur **GitHub**.

Instructions

1. Création et navigation dans le projet (Bash)

1. Créez un répertoire nommé `mon_projet_git`.
2. Accédez à ce répertoire.
3. Créez un fichier texte `notes.txt` contenant la phrase suivante :

```
Mon premier fichier versionn avec Git
```
4. Affichez le contenu du fichier pour vérifier son contenu.

2. Initialisation du dépôt Git et premier commit

1. Initialisez un dépôt Git dans ce répertoire.

2. Vérifiez l'état du dépôt.
3. Ajoutez `notes.txt` à la zone de staging.
4. Effectuez un commit avec le message "Ajout du fichier `notes.txt`".

3. Création d'une branche et modification du fichier

1. Créez une nouvelle branche appelée `dev`.
2. Basculez sur la branche `dev`.
3. Modifiez `notes.txt` en ajoutant une nouvelle ligne :

```
Modification sur la branche dev
```
4. Vérifiez les changements.
5. Ajoutez et committez ces modifications avec le message "Mise à jour du fichier `notes.txt` sur `dev`".

4. Fusionner la branche `dev` dans `main`

1. Retournez sur la branche `main`.
2. Fusionnez la branche `dev` dans `main`.
3. Supprimez la branche `dev` après la fusion.

5. Publier le projet sur GitHub

1. Créez un **nouveau dépôt** sur **GitHub** nommé `mon_projet_git`.
2. Ajoutez ce dépôt distant à votre projet local.
3. Poussez la branche `main` vers GitHub.

Bonus (Optionnel) : Clonage et collaboration

1. Clonez ce projet dans un autre dossier pour simuler une autre machine.
2. Modifiez le fichier `notes.txt`, faites un commit et poussez les changements sur GitHub.

Indications

- Utilisez la commande `git status` pour voir les changements à chaque étape.
- Vérifiez votre historique Git avec `git log -oneline`.