Feuille d'exercices Git

Noter dans le tableau final, pour chaque question la commande que vous avez lancé si vous avez utilisé **git**.

1- Créer un nouveau dépôt git

Munissez vous de la feuille "mémento des commandes git". Déplacez vous avec un terminal dans le dossier des exercices Python. Nous allons utilisez ce code comme un projet logiciel à suivre à l'aide de git.

1. Créer un nouveau dépôt à cet endroit avec **git init.**
2. Afficher l'état du dépôt. Que signifie non suivi ?
3. Afficher la liste des commits du dépôt pour constater qu'il n'y en a aucun.
4. Essayer de créer un commit directement avec **git commit -m.** Pourquoi cela échoue-t-il ?
5. Ajouter tous les fichiers au suivi. Vérifier l'état avec **status**. L'ajout n'est pas définitif tant qu'il n'a pas été validé par un commit.
6. Désinclure (unstage) un des fichiers python (qui ne sera donc pas inclus dans le prochain commit). Vérifier le résultat avec **status**.
7. Vérifier le code ajouté avec **diff**.
8. Valider ces modifications en créant un commit avec le message "version initiale"
9. Ajouter le fichier restant supprimé précédemment et créer un autre commit avec le message "fichier oublié".
10. Modifier le message du dernier commit en "fichier restant".
11. Lancer VSCode. Ajouter le message `# une ligne supplémentaire` à la fin de deux des fichiers python. Que constatez vous dans la vue **Explorer** ? Dans la vue **Source Control** ?
12. Ajouter les nouvelles modifications au suivi dans la vue **source control** et créer un nouveau commit "test VSCode".

2- Explorer un dépôt git

1. Cloner le dépôt https://github.com/miguelgrinberg/microblog

Il s'agit d'une application Flask assez complète et complexe que Miguel Grindberg vous propose de coder dans son Méga tutorial : https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world. Suivre ce tutorial est un très bon exercice pour progresser en python (un peu plus dur

que ce qu'on fera ici mais très bien expliqué).

2. Lancer **git pull** et **git status.** Que remarquez vous ?
3. Afficher la liste des commits du dépôt. Combien y a-t-il de chapitres ? Combien y a-t-il de commit en tout ?
4. Installer **tig** et explorez le dépot avec. Utiliser les flèches et Entrer pour naviguer, Q pour quitter **tig**.
5. Avec tig identifier le numéro du commit de la fin du premier chapitre dans lequel le fichier **routes.py** est créé.
6. Lancer VSCode. Installer l'extension GitLens. Aller dans la vue GitLens
7. Observer l'historique du dépôt. On y retrouve la liste des commit et les modifications de chaque commit comme dans tig.
8. Ouvrir le fichier **routes.py**. Observer l'historique du fichier.
9. Placer vous au niveau du commit ou **routes.py** est créé avec **git checkout <num_commit>**. **HEAD** pointe désormais sur ce commit. Le dépôt est dans un état "HEAD détachée". Git status permet de le revérifier.
10. Utiliser **git reflog**. Que remarque-t-on ? **git reflog** peut vous sauver la mise si vous ne comprenez pas dans quel état est votre dépôt.
11. Replacer **HEAD** au niveau du dernier commit de **master** avec **git checkout …** Tout l'historique apparaît à nouveau.
12. Où sont caché les fichier dans git ? Comment supprimer le suivi git d'un dépôt ?

3- Branches et Merges

Reprenons les exercices **Python** en utilisant des branches **Git:**

1. Aller dans le dossier d'exercices python. Identifier le numéro de la question suivante à faire.
2. Lister les branches.
3. Vérifier que vous êtes bien sur la branche **master** et que toutes les modifications on été validées/commit.
4. Créer une nouvelle branche avec le nom "question <num_question>" ou num_question est le numéro de la prochaine question à faire (**git checkout -b …**)**.**
5. Lancer **git status** pour confirmer que vous avez bien basculé sur la bonne branche.
6. Lister à nouveau les branches.
7. Faire l'exercice de la question (voir feuille exercice python). Vérifier avec **git diff** que git

identifie bien vos modifications.

8. Tester le code en lançant le programme. Et corriger si nécessaire.
9. Une fois que vous êtes satisfait.e du résultat : Inclure le(s) fichier(s) que que vous avez modifié.
10. A l'aide de la vue **Source Control** de **VSCode** relisez vos modifications.
11. Créez un nouveau commit.
12. Dans la vue **GitLens**, chercher la zone qui liste les branches et vérifier que votre nouveau commit est bien là ou il doit être.

Nous allons faire un merge de cette branche dans master !

13. Basculer sur la branche **master** à l'aide de **git checkout …** .
14. Lancer **git merge <nom_de_votre_branche> .** Git vous indique s'il y a des conflits et dans quel fichiers. Normalement il n'y a pas de conflit pour le moment

Nous allons maintenant créer un dépôt dans la forge framagit.

15. Créer un compte framagit.
16. Créer un nouveau projet "Exercices python". (privé ou public selon votre convenance)
17. Suivre les indications du site pour ajouter ce projet comme remote python de votre dépôt local.
18. Pousser le code de vos exercices sur framagit.

Question	Commande