Contrôle Continu – GitHub.

Vous répondrez à chacune des questions sur ce fichier texte.

Pour chacune d'entre elles, vous effectuerez la tache demandée et vous écrirez la ou les commandes nécessaires sous la question.

1. A l'aide du terminal, créez un dossier "CC\_nom\_prenom" dans lequel vous initialiserez un repository git.

QUESTIONS DE COURS.

2. Qu'est ce que Git et à quoi sert-il ?

R : Git est un logiciel de gestion de versions. Il permet de stocker différentes versions du même travail et donc de suivre étape par étape les différents changements dans le projet

3. Comparez Git et GitHub.

R : Git est un logiciel local alors que Github est plutôt un hébergeur distant qui offre une interface et stocke les différentes versions enregistrées par Git. Github permet un travail collaboratif plus aisé étant donné que différentes machines peuvent être connecté au même projet et inversement

4. Donnez les deux espaces dans lesquels peuvent se trouver les fichiers avant de réaliser un commit.

R : Les fichiers peuvent se trouver soit dans le repository local ou dans la staging area

QUESTIONS PRATIQUES.

5. Après avoir répondu aux questions de cours, ajoutez ce fichier dans votre repo et faites votre premier commit.

R: - git add controle\_continu.docx

* git commit -m “first commit controle continu”

6. Créez une branche feature et placez vous dessus. Donnez les 3 méthodes de création de branche.   
R: - git checkout -b feature

* git branch feature
* git switch -c feature

7. Ajoutez un fichier texte film.txt à votre dossier et écrivez votre film préféré à l'intérieur.

R: echo "twilight" > film.txt

8. Ajoutez un fichier texte serie.txt à votre dossier et écrivez votre série préférée à l'intérieur.

R: echo "scandal" > serie.txt

9. Faites un commit en ajoutant uniquement le fichier film.txt à votre commit.

R : - git add film.txt

* git commit -m “film prefere”

10. Ajoutez un fichier citation.txt dans lequel vous écrirez votre citation préférée.

R : echo "Great things take time" > citation.txt

11. Faites un commit qui ajoute toutes les modifications.

R: - git add .

* git commit -m “film, serie et citation pref”

12. Sur une nouvelle branche add\_images, créez un fichier "ignore.txt" et ignorez le. Vous ferez ensuite un capture d'écran qui montre que le

fichier est bien ignoré et vous ajouterez cette capture dans un dossier "images". Vous pourrez ensuite commit les changements.

R: - git switch master

* git switch -c add\_images
* touch ignore.txt
* echo “ignore.txt” > .gitignore

13. Faites une capture d'écran de vos branches et ajoutez la au dossier image et faites un commit.

R: - git add .

* git commit -m "add gitgnore image proof"

14. Mergez vos branches feature et add\_images dans master.

R: - git switch master

* git merge feature
* git merge add\_images

15. Faites une capture d'écran de l'historique de vos commit sur master, ajoutez la au dossier image et faites un commit.

R: - git add .

* git commit -m "add master git log after merge"

16. Créez un repo en ligne dans lequel vous pousserez votre dossier.

R: - git remote add origin <https://github.com/soraya-21/tp_github_controle_continu.git>

* git push origin master

17. Sur GitHub, créez un nouveau fichier origin.txt sur la branche master. Puis tirez le dans votre repo local.

R: - git pull origin master

18. Créez une nouvelle branche feedback, ajoutez un fichier feedback.txt dans lequel vous écrirez votre feedback sur le cours

(vous ne serez ni avantagé ni pénalisé selon le feedback). Créez ensuite une Pull Request et mergez la dans master depuis GitHub.

R: - git switch -c feedback

* touch feedback.txt
* git add feedback.txt
* git commit -m "feedback.txt"
* git push origin feedback

Lien de votre repo GitHub : https://github.com/soraya-21/tp\_github\_controle\_continu.git

Vous laisserez votre repo distant en public et écrirez le lien dans ce fichier que vous rendez sur classroom.