

Contrôle Continu - GitHub.

Vous répondrez à chacune des questions sur ce fichier texte.
Pour chacune d'entre elles, vous effectuerez la tâche 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.

2. Git est un outil qui permet de suivre les changements dans des fichiers, surtout dans le code d'un projet. Il est utilisé par les développeurs pour gérer les différentes versions de leurs projets et pour travailler en équipe

QUESTIONS DE COURS.

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

3. Comparez Git et GitHub. 3. Git est l'outil, tandis que GitHub est un service qui utilise Git pour faciliter le partage et la collaboration

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

4. Le répertoire de travail et la zone de transit

QUESTIONS PRATIQUES.

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

6. Créez une branche feature et placez vous dessus. Donnez les 3 méthodes de création de branche.

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

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

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

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

11. Faites un commit qui ajoute toutes les modifications.

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.

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

14. Mergez vos branches feature et add_images dans master.

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

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

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

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

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

Lien de votre repo GitHub :

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