Questionnaire fin de module Git et github
(temps 1h00)

1. Quelle est l'utilité de git ?
   Git mémorise toutes les créations, modifications ou suppressions des différents fichiers d'un projet afin de permettre de retracer toute l'évolution du projet, fichier par fichier. Le but est de pouvoir à tout moment restaurer l'une de ces versions (versionner).

2. Dans quelle situation est-il nécessaire d'utiliser git ?
   Pour les projets de groupe et en entreprise car git permet facilement de mettre le code en commun et donc de gérer simplement le travail en équipe.

3. Quelle commande permet d'initialiser git dans un projet ?

   git init

4. Où est-ce que la commande git init doit être exécutée ?

   A la racine du projet pour pouvoir enregistrer chaque création, modification ou suppression de fichiers à l'intérieur de ce répertoire.

5. Combien d'étapes sont nécessaires pour enregistrer correctement un fichier via git ?

   2 étapes : l'ajout des fichiers (git add) à sauvegarder et le commit (git commit).

6. Quelle commande permet d'ajouter l'ensemble des fichiers modifiés ?

   git add .

7. Le code suivant est-il correct ? `git commit`

Non, il faut toujours ajouter un message au commit (git commit -m "first commit).

8. Écrire le code pour enregistrer le fichier “script.js” via git

   git add script.js git commit -m “commentaire”

9. Quelle commande permet d’afficher l’historique des versions?

   git log

10. Écrire toutes les commandes qui permettent de créer un nouveau projet, d’initialiser git, de créer un premier fichier script.js et de l’enregistrer dans le dépôt :

    mkdir newproject //création d’un nouveaau projet

    cd newproject // déplacement dans le dossier crée

    git init // Initialisation de git sur le dossier

    touch script.js // création du fichier

    git add script.js // ajout du fichier

    git commit -m “fichier enregistré” // prise de photo

11. Quelle est l'utilité des branches ?

    Les branches nous permettent de travailler sur différentes versions de notre projet sans prendre le risque de travailler sur une seule version.

12. Quelle commande permet de créer une branche ?

    git branch newBranche

13. Quelle commande permet de passer d'une branche à une autre ?

git checkout newBranche

14. Que fait la commande git branch ?

Cette commande permet de lister l'ensemble des branches existantes sur notre projet.

15. Quelles sont les étapes pour fusionner le code de deux branches ?

git checkout master // permet d'aller sur la branche master

git merge newBracnhe // Fusionner les branches

git branch -d newBranche // permet de supprimer la branche

16. Quelle est la différence entre dépôt local et dépôt distant ?

Le dépôt local est stocké sur l'ordinateur alors que le dépôt distant est en ligne et peut être accessible par d'autres personnes, via GitHub par exemple.

17. Quelle commande permet d'associer notre dépôt local à un dépôt distant ?

git remote add origin https://urlDuDepot.git

18. Quelle commande permet d'envoyer les modifications sur le dépôt distant ?

git push origin master

19. Quelle commande doit être effectuée avant un git push pour recevoir les modifications du dépôt distant ?

git pull origin master

20. **Bonus** : Écrire toutes les commandes qui permettent de cloner un dépôt à l'url suivant https://github.com/jquery/jquery.git puis d'y associer un nouveau dépôt distant

    git clone https://github.com/jquery/jquery.git // Cloner le projet en local

    git remote add origin https://github.com/acebroke/projet_lundi.git // Pour lier le dépôt distant et le dépôt local