

1/- C'est quoi Git ?

Git est un logiciel de gestion de versions (Version Control System) qui suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans rien écraser.

Cela permet de retrouver les différentes versions d'un fichier ou d'un lot de fichiers connexes et ainsi éviter des problèmes, Avec Git, on peut par exemple retrouver sans problème la version qui fonctionnait en cas de problème.

2/- C'est quoi un repository git ?

C'est un répertoire de notre système de fichier dans lequel il y a un répertoire. Git ou on aura toute l'historique des modifications pour créer un repo git on utilise la commande git init

3/- Quelle commande faut t-il écrire pour valider un ajout dans un répertoire git ?

On utilise git add .

4/- que fait la commande git push ?

Elle permet d'envoyer ses commits vers une branche distante choisie

5/- quel est l'avantage de git par rapport à svn?

Vous n'avez pas besoin d'une connectivité au réseau de manière continue par exemple pour pouvoir travailler sur votre projet de n'importe où mais aussi avec git on a la possibilité

Git est décentralisé

La protection en cas de panne ou de perte du dépôt principal

6/- c'est quoi la commande git stash ?

Git stash est une commande de Git qui permet de "mettre de côté" des modifications. Cette fonctionnalité est très pratique, notamment lorsque le développeur a besoin de changer de branche rapidement.

Cette fonctionnalité lui permet de ne pas se presser à effectuer un commit bâcler pour pouvoir changer de branche.

Git stash va récupérer les modifications en cours et les enregistrer dans un conteneur qu'on appellera un stash.

7/- c'est quoi la commande git branch ?

Cela permet de créer une nouvelle branch exemple « git branch nom_de_branch » ou de lister les branches existantes exemple « git branch -a »

8/- c'est quoi la commande git clone ?

Cela permet de récupérer un repository sur Github

9/-c'est quoi la commande git config ?

Cela permet de paramétrer git (nom prenom email ...)

10/- c'est quoi la commande git init ?

Cela permet d'initialiser un répertoire local en repository git

11/- c'est quoi un conflit git ?

Un conflit intervient quand des fichiers du même noms et qui n'ont pas les mêmes données

12/- comment résoudre un conflit git ?

En utilisant git slash ou en faisant la correction des lignes manuellement

13/- c'est quoi la commande git remote ?

git remote relie un repo local un repo distant sur le github

14/- quelle est la différence entre git remote et git clone ?

Git clone permet de récupérer un repository du github en local

Git remote permet de relier un repository local a un repo distant

15/- c'est quoi la commande git status ?

Cela Permet de connaître le statut des fichiers (local en rouge, en cours d'envoi en vert et envoyé n'affiche rien)

16/- c'est quoi la commande git checkout ?

Cela permet de changer de branch ou de version de commit

17/- c'est quoi la commande git log ?

Cela affiche les commits et leur hashcode on peut utiliser ces derniers apres un git checkout afin de naviguer entre les différentes version

18/- donner la suite de commandes à écrire pour envoyer un fichier sur un repository distant

Git init

Git remote add origin lien git

Git pull

Git add

Git push

19/- c'est quoi github ?

C'est un service de repository distant et communautaire

20/- citer 3 avantages d'utiliser git dans un projet

La possibilité de revenir a n'importe quel moment à n'importe quelle version du projet

La possibilité de travailler sur un projet à plusieurs et à distance

La possibilité de l'utiliser en local et hors connexion

21/- c'est quoi le fichier .gitignore

C'est un fichier qui permet de filtrer les fichiers qu'on push afin d'optimiser l'envoi et d'éviter d'écraser des fichiers de réglage qui sont propre a chaque développeur

22/- quelles sont les étapes permettant de lier un projet git local à github?

Créer un repo sur github

Cloner le repo

Deposer notre projet dans le repo cloner

Faire un commit

Faire un push

23/- quel est le role des branches sur git ?

Une branche correspond à une version parallèle de celle en cours de développement. Une branche peut servir à développer de nouvelles fonctionnalités ou encore à corriger d'éventuels bugs sans pour autant intégrer ces modifications à la version principale du logiciel ou de l'application.

Les branches permettent de segmenter différentes versions en cours de développement.

En règle générale, la branche nommée *Master* correspond à la branche principale du projet.