



Accueil > Cours > Gérez du code avec Git et GitHub > Quiz : Testez ce que vous avez appris

Gérez du code avec Git et GitHub

6 heures  Facile

Mis à jour le 12/03/2021



Testez ce que vous avez appris

Vous pouvez repasser ce quiz dès maintenant.

Si vous échouez, vous pourrez repasser ce quiz dans 24h.

Soyez vigilant sur vos réponses !

[REPASSER CE QUIZ](#)

Compétences évaluées



Utiliser les commandes de base de Git

Question 1

Quelles sont les trois zones locales majeures dans Git ?

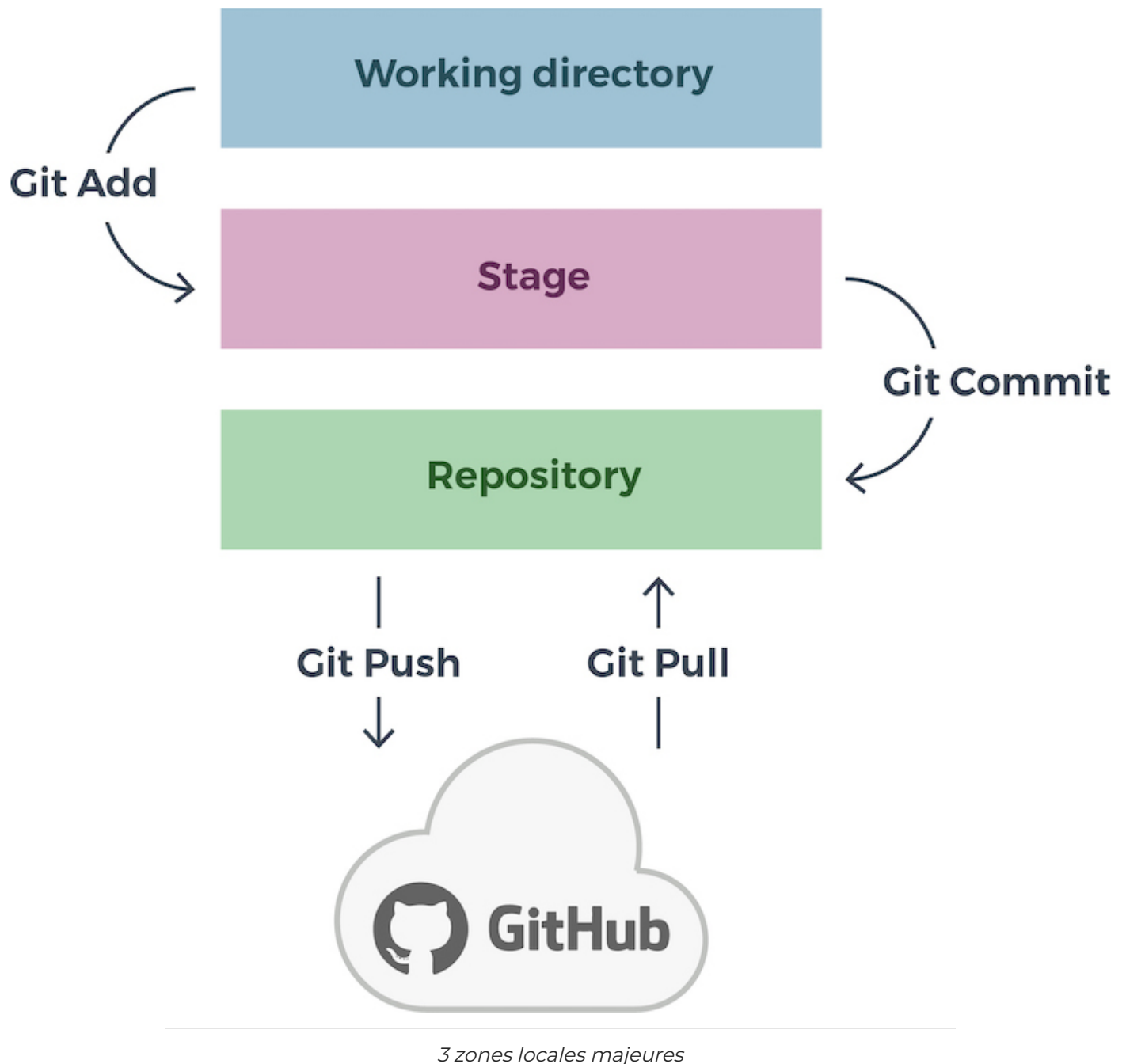
Attention, plusieurs réponses sont possibles.

✓ ☒ Working directory

☐ Cloud

✓ ☒ Repository

✓ ☒ Index



Git gère les versions de vos travaux locaux à travers 3 zones locales majeures :

- **le répertoire de travail** (working directory/WD) ;
- **l'index ou stage** ;
- **le dépôt local** (Git directory/repository).

Le répertoire de travail désigne **l'arborescence** de vos fichiers, c'est-à-dire tous vos fichiers et répertoires qui sont indépendants de Git. Ils sont même là avant que vous réalisiez votre git init ! L'index ou stage désigne tous les fichiers modifiés que vous souhaitez voir apparaître dans votre **prochain commit**. C'est avec la commande `git add` que l'on ajoute un fichier à l'index. Le dépôt local est **l'historique de l'ensemble de vos actions** (commits, configurations...).

Question 2

Vous avez créé votre dépôt distant sur GitHub, que l'on appellera monDepotDistant, et vous souhaitez maintenant le lier avec votre dépôt local sur Git. DEP représente le nom court qui sera utilisé pour appeler le dépôt.

Quelle ligne de commande devez-vous taper ?

☐

```
git remote DEP https://github.com/xxxxxx/monDepotDistant
```

☐

```
git remote DEP add  
https://github.com/xxxxxx/monDepotDistant
```



```
git remote add DEP  
https://github.com/xxxxxx/monDepotDistant
```

☐

```
git add DEP  
https://github.com/xxxxxx/monDepotDistant
```

Pour lier votre dépôt local au dépôt distant, vous devez effectuer la commande :

```
git remote add DEP  
https://github.com/xxxxxx/monDepotDistant
```

La commande git remote doit être effectuée sous la forme : git remote add leNomCourt "URL de votre dépôt".

Les autres commandes ne sont pas valides.

Question 3

Maintenant que votre dépôt distant est lié à votre dépôt local, vous allez devoir dupliquer son contenu en local. Pour ce faire, nous devons utiliser :

☐

```
git clone DEP  
https://github.com/xxxxxx/monDepotDistant
```

☐

```
git duplicate  
https://github.com/xxxxxx/monDepotDistant
```



```
git clone  
https://github.com/xxxxxx/monDepotDistant
```

- ☐ `git duplicate DEP`

La commande `git clone` permet de dupliquer le contenu distant en local. Pour être valide, cette commande doit être écrite sous la forme : `git clone https://github.com/nom-dutilisateur/monDossier`. La commande `git duplicate` n'existe pas.

Question 4

Nous allons désormais ajouter des branches à notre dépôt. Je souhaite créer trois nouvelles branches : "Branch1", "Branch2" et "Branch3", et basculer sur ma branche "Branch2". Que dois-je faire ?

- ☐ `git branch`
`git branch`
`git branch`
`git checkout branch`



`git branch Branch1`
`git branch Branch2`
`git branch Branch3`
`git checkout Branch2`

- ☐ `git checkout Branch1`
`git checkout Branch2`
`git checkout Branch3`
`git branch`

Pour créer chacune des branches, vous devez utiliser la commande `git branch`, puis réaliser un `git checkout` sur la branche où vous souhaitez basculer. Utilisée seule, la commande `git branch` permet de visualiser les différentes branches d'un projet.

Question 5

Quelle commande permet de consulter nos branches, et quel est le résultat de cette commande ?



`git branch`
main
Branch1
*Branch2
Branch3



```
git checkout  
main  
Branch1  
*Branch2  
Branch3
```



```
git branch  
Branch1  
*Branch2  
Branch3
```



```
git checkout  
Branch1  
*Branch2  
Branch3
```

La commande `git branch` permet de consulter l'état de notre arbre, et ici de visualiser 4 branches : `Branch1`, `Branch2`, `Branch3` et notre branche principale, la branche "main". La commande `git checkout` permet de basculer d'une branche à une autre.

Question 6

Vous devez maintenant réaliser des modifications sur votre branche `Branch2`, puis créer une version en indiquant le message "Je suis le roi du commit". Comment faites-vous cela ?



```
git commit "Je suis le roi du commit"
```



```
git commit -m "Je suis le roi du commit"
```



```
git commit -m -n "Je suis le roi du commit"
```



```
commit
```

`git commit -m` permet de réaliser un commit en indiquant un message de description. Les autres commandes ne sont pas valides.

Question 7

Dans quel ordre dois-je réaliser les 3 commandes ?



```
git add fichier.html
git commit -m "message"
git push origin main
```



```
git add fichier.html
git push origin main
git commit -m "message"
```



```
git push origin main
git add fichier.html
git commit -m "message"
```



```
git commit -m "message"
git add fichier.html
git push origin main
```

Pour commencer, on indexe le fichier .html avec la commande `git add`, puis on crée une version avec la commande `git commit`, et enfin on envoie cette version sur le dépôt distant avec la commande `git push origin main`. Les autres commandes ne sont pas valides.

Question 8

Je veux fusionner la branche "modifications" à la branche principale du projet. Quelles commandes dois-je réaliser ?



```
git checkout main
git merge modifications
```



```
git checkout modifications
git merge main
```



```
git merge modifications
git checkout main
```

○ `git merge main`
`git checkout modifications`

La commande merge doit s'utiliser à partir de la branche dans laquelle nous voulons apporter les évolutions. Nous voulons apporter les évolutions de la branche "modifications" dans la branche principale (main). Nous devons donc utiliser les commandes :

```
git checkout main
git merge modifications
```



**APPREHENDEZ LE SYSTÈME DE
BRANCHES**

**PRATIQUEZ ET CORRIGEZ VOS ERREURS
SUR UN DÉPÔT LOCAL**



Les professeurs

Luc Bourrat

Mentor et développeur web

Tiffany Lestroubac

Développeur Fullstack / Mentor / Evaluatrice / Rédactrice

Mila Paul

I am an academic instructor of Computer Science and Information Security, freelance Blockchain developer, and Cyber Operations SME!

OPENCCLASSROOMS



ENTREPRISES



CONTACT



EN PLUS



