Utilisez Git et GitHub pour vos projets de développement

12 heures Facile

Mis à jour le 11/03/2021

Réparez les erreurs les plus courantes

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées

Corriger les erreurs courantes sur GitHub

Question 1

Maintenant que je suis sur ma Branch1, nous allons ajouter un fichier "File2.txt".

Finalement, je ne veux plus ajouter mon fichier sur cette branche mais sur une nouvelle branche, "BranchFile". Que dois-je faire ?

- ```
  git status
  git branch BranchFile
  git checkout BranchFile
  git status apply
  ```

- ```
  git status
  git stash
  git branch BranchFile
  git checkout BranchFile
  git status apply
  ```

✓ ◉
```
git status
git stash
git branch BranchFile
git checkout BranchFile
git stash apply
```

Nous abordons ici la remise. Nous allons devoir remiser les changements faits sur Branch1 pour les appliquer ensuite sur la nouvelle branche, "BranchFile". `git stash` *permet de remiser des modifications et* `git stash apply` *permet d'appliquer une remise sur une branche.*

Question 2

```
git commit --amend -m "Test"
```

Cette commande va :

- ○ permettre de revenir au commit précédent
- ✓ ◉ permettre de modifier le message du commit précédent
- ○ permettre de créer un nouveau commit avec le message "Test"

git commit --amend -m modifie le message du commit précédent. C'est le -m qui indique que nous allons modifier le message.

Question 3

Je souhaite revenir sur le commit précédent et modifier son contenu en ajoutant un fichier, que dois-je faire ?

○
```
git commit --amend --no edit
git add monFichierOublié
git commit
```

✓ ○
```
git add monFichierOublié
git commit --amend --no-edit
```

✗ ◉
```
git commit --amend
git add monFichierOublié
git commit
```

○

```
git add monFichierOublié
git commit --amend
```

`git add monFichierOublié` *permet d'ajouter mon fichier.*
`git commit --amend --no-edit` *permet de modifier le commit sans changer le message.*

Question 4

Quelle est la différence entre Git revert et Git reset ?

- ✗ (●) Git revert ne va réinitialiser qu'un commit alors que Git reset réinitialise tout.
- ✓ ( ) Git revert va créer un nouveau commit alors que Git reset, non.
- ( ) Git reset va créer un nouveau commit alors que Git revert, non.

Git reset ne crée jamais un nouveau commit alors que Git revert, oui.

Question 5

Je souhaite savoir qui a touché à une ligne en particulier dans le fichier test.html, quelle commande vais-je devoir exécuter ?

- ( )
```
git log
```
- ( )
```
git reflog
```
- ✓ (●)
```
git blame
```
- ( )
```
git cherry pick
```

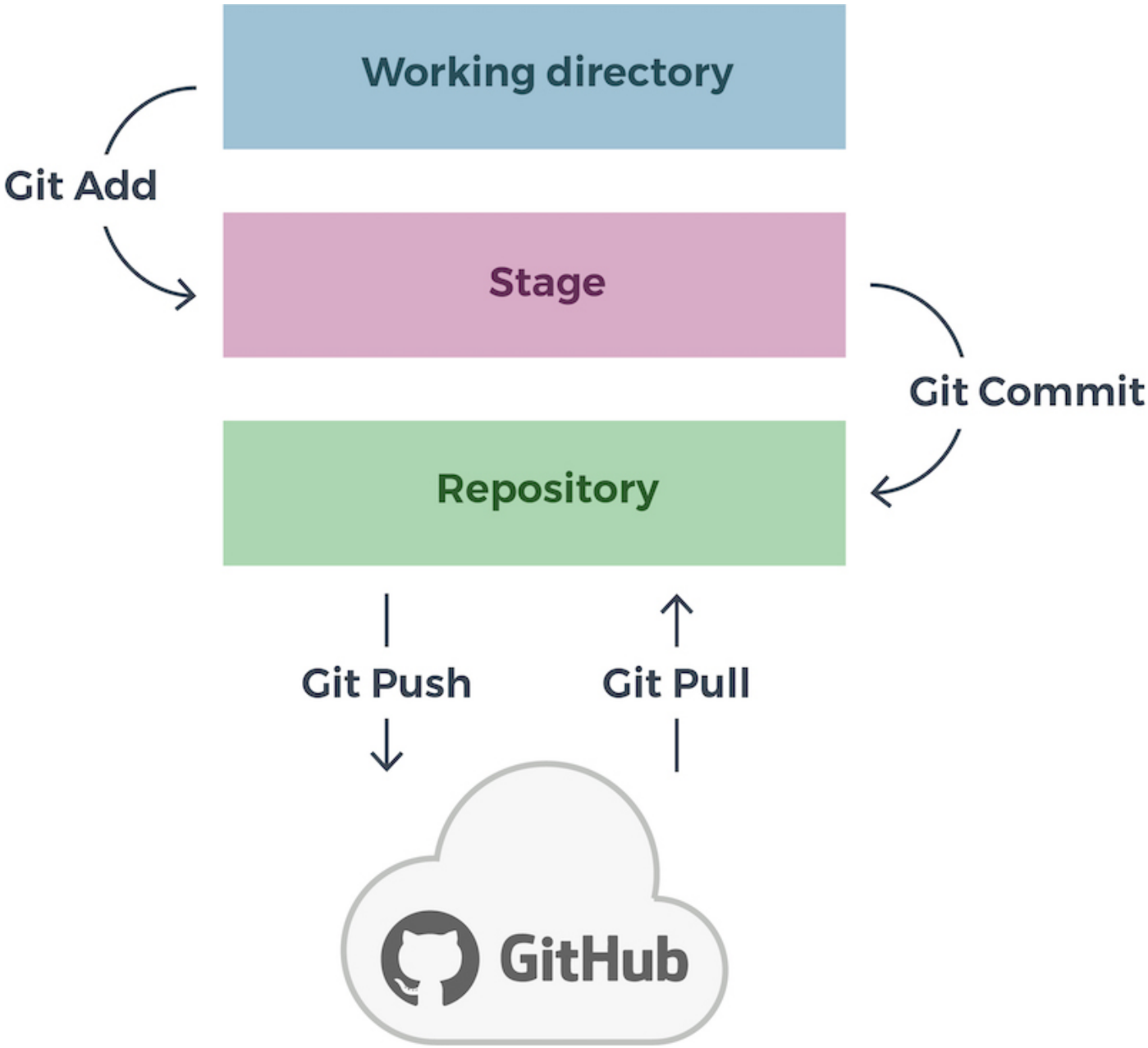
Si vous ne vous en souvenez plus, allez faire un petit tour sur le cours "Qui s'est amusé dans mon dépôt ? Git blame".

Question 6

Quelles sont les trois zones locales majeures ?

Attention, plusieurs réponses sont possibles.

- [x] Working directory
- [ ] Cloud
- [x] Repository
- [x] Stage



3 zones locales majeures

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*** *(nous préférerons le second terme) ;*
- ***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 fonction* `git add` *que l'on ajoute un fichier au stage.*

*Le dépôt local est **l'historique de l'ensemble de vos actions** (commits, configurations...).*

Question 7

Qu'est ce qu'un SHA ?

- ○ Un SHA est un numéro aléatoire permettant de se connecter à GitHub
- ✓ ◉ Un SHA est un identifiant pour les commits et autres actions gardés en mémoire par Git
- ○ Un SHA est un petit animal qui ronronne

Le SHA, c'est ce grand code qui vous permettra de revenir à un commit exact. C'est l'identifiant de votre action !

CORRIGEZ UN COMMIT RATÉ

IDENTIFIEZ LA STRUCTURE DE FICHIER DE GIT

Les professeurs

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!