

Slide 1

Comment inviter des collaborateurs sur le dépôt

Points clés

Ce diaporama explique comment le propriétaire d'un dépôt peut inviter d'autres collaborateurs à travailler sur le projet. Nous verrons également la différence entre `git pull` et `git clone`, ce qu'est une branche Git, quelle est la branche principale de notre projet et son importance, et enfin comment manipuler les branches (créer, basculer, supprimer, fusionner, etc.).

Notes du présentateur

Dans cette présentation, nous allons aborder la manière dont le propriétaire d'un dépôt peut inviter d'autres collaborateurs à rejoindre le projet. Nous verrons également la différence entre `git pull` et `git clone`, ce qu'est une branche, quelle est la branche principale de notre projet Git et son importance, et enfin comment manipuler les branches.

Slide 2

Invitation des collaborateurs

- Le propriétaire du dépôt peut envoyer une invitation par e-mail.
- L'invitation contiendra un lien que le collaborateur devra utiliser pour rejoindre le dépôt.
- Le propriétaire peut également accorder un accès direct au dépôt au collaborateur.

Notes du présentateur

Le propriétaire du dépôt peut inviter d'autres collaborateurs à travailler sur le projet en leur envoyant une invitation par e-mail. L'invitation contiendra un lien que le collaborateur devra utiliser pour rejoindre le dépôt. Le propriétaire du dépôt peut également donner un accès direct au dépôt au collaborateur.

Slide 3

Différence entre `git pull` et `git clone`

- **git clone** permet de créer une copie locale d'un dépôt distant.
- **git pull** permet de récupérer les modifications les plus récentes du dépôt distant et de les fusionner dans votre branche locale.

Notes du présentateur

La commande `git clone` est utilisée pour créer une copie locale d'un dépôt distant. La commande `git pull` permet quant à elle de récupérer les modifications les plus récentes apportées sur le dépôt distant et de les fusionner dans votre branche locale.

Slide 4

Qu'est-ce qu'une branche Git ?

[Image of Un arbre avec des branches différentes]

- Une branche Git est une ligne de développement distincte au sein d'un dépôt.
- On utilise les branches pour travailler sur de nouvelles fonctionnalités ou corriger des bogues sans affecter la branche principale.
- Chaque branche possède son propre historique de commits.

Notes du présentateur

Une branche Git est une ligne de développement indépendante au sein d'un dépôt. Les branches permettent de travailler sur de nouvelles fonctionnalités ou de corriger des bogues sans affecter la branche principale. Chaque branche possède son propre historique de commits, retraçant l'évolution du code qui lui est associé.

Slide 5

La branche principale

[Image of Un arbre avec une branche surlignée]

- La branche principale est la branche par défaut dans un dépôt Git.
- C'est généralement la branche qui est déployée en production.
- La branche principale est souvent nommée "main" ou "master".

Notes du présentateur

La branche principale est la branche par défaut dans un dépôt Git. C'est généralement la branche qui est déployée sur les environnements de production. La branche principale porte souvent le nom de "main" ou "master".

Slide 6

Manipulation des branches

- **git branch** : Lister les branches du dépôt
- **git checkout** : Basculer vers une branche spécifique
- **git branch -d** : Supprimer une branche
- **git merge** : Fusionner une branche dans une autre
- **git rebase** : Rebaser une branche sur une autre branche

Notes du présentateur

- `git branch` permet de lister les branches présentes dans le dépôt.
- `git checkout` permet de basculer vers une branche spécifique pour y travailler.
- `git branch -d` permet de supprimer une branche.
- `git merge` permet de fusionner une branche dans une autre branche.
- `git rebase` permet de rebaser une branche sur une autre branche.

J'espère que ces diapositives vous seront utiles ! N'hésitez pas à me poser des questions si vous en avez.