

Utilization de GIT

Git est un outil de versionnage de code source. Il permet de garder un historique des différentes versions de votre code source. Plus tard, nous l'utiliserons également comme outil de collaboration.

1. Ouvrez un terminal et rendez vous dans votre répertoire de votre projet (td00/src) et transformez le en dépôt Git en utilisant la commande `git init`.
Attention : soyez très attentif avec les commandes Git, certaines erreurs peuvent être compliquées à résoudre. Pour cette question, soyez bien sûr d'être dans le répertoire de le projet (td00/src).
3. Ajoutez les fichiers sources du projet avec la commande `git add fichier1` (ici par exemple Point.java). Vous pouvez répéter cette commande autant de fois que nécessaire.
4. Effectuez votre premier commit en utilisant la commande `git commit`. Le commit va enregistrer les sources de votre projet telles qu'elles sont actuellement en tant que première version de votre projet (`git commit -m "git message"`).
5. Apportez une modification quelconque au fichier point.java (par exemple, créez un autre constructeur avec le paramètre Point p, sauvegardez le fichier, et rendez vous dans le terminal et tapez `git diff`. Qu'observez vous ?
6. Ajoutez la modification apportée en tapant de nouveau `git add` suivi du chemin vers le fichier point.java. Et effectuez votre deuxième commit en tapant `git commit`. Entrez un nouveau message pour décrire les changements apportés (par exemple : ajouter un nouveau constructeur avec un paramètre Point p).
7. Consultez l'historique des modifications apportées à votre projet depuis la version initiale en utilisant la commande `git log`. Vous pouvez également utiliser l'interface graphique `gitk` pour visualiser l'historique de votre projet. Note : si vous tapez `gitk` dans le terminal, vous n'aurez plus la main sur le terminal jusqu'à la fin de l'exécution de la commande `gitk` (i.e. jusqu'à ce que vous fermiez l'interface graphique). Pour lancer `gitk` tout en gardant la main sur le terminal, vous devez taper la commande `gitk &`.
8. Chaque commit est identifié par un hash SHA-1 représenté sous la forme d'une séquence de caractères alpha-numériques. A l'aide de la commande `git log`, trouvez et copiez le hash de votre premier commit (celui que vous avez normalement écrit par le message initial import).

9. En utilisant `git diff hash-commit-1 hash-commit-2`, vous pouvez déterminer la différence apportée entre deux commits. Testez cette commande pour visualiser la différence entre le premier et le deuxième commit.

10. Pour faciliter la tâche, le commit le plus récent est toujours référencé par le mot clé `HEAD`. Il est également possible de faire référence au commit précédent le dernier commit grâce au mot clé `HEAD^`. Reproduisez le résultat de la question précédente sans utiliser les hash des commits.

Vous connaissez les bases de l'utilisation de git en local. Vous êtes très fortement encouragés à suivre le tutoriel proposé par les créateurs de git disponible ici <https://git-scm.com/docs/gittutorial>. (A la maison.)