

Guide Étape par Étape pour Utiliser Git

Introduction

Ce guide explique comment utiliser Git dans le cadre de projets de programmation. Nous allons aborder les étapes pour cloner un dépôt, créer des dossiers et des fichiers pour des exercices spécifiques, ajouter ces fichiers à la staging area, les committer et enfin les pousser sur le dépôt distant. Chaque étape sera accompagnée d'explications détaillées pour vous aider à comprendre le rôle de chaque commande.

1. Cloner le Dépôt

Cloner le dépôt dans votre espace de travail:

```
git clone https://votre_lien_git.git
cd nom_du_dépôt
```

Explication:

`git clone` : Cette commande crée une copie locale d'un dépôt Git distant. Cela vous permet de travailler sur le projet sur votre machine locale.

`cd nom_du_dépôt` : Change le répertoire courant pour entrer dans le répertoire du projet cloné.

2. Créer des Dossiers et Fichiers pour les Exercices

Vous pouvez maintenant créer des dossiers pour vos exercices à l'intérieur du répertoire cloné. Par exemple, pour créer les dossiers `ex01` et `ex02`:

```
mkdir ex01
mkdir ex02
```

Ensuite, vous pouvez créer des fichiers dans ces dossiers. Par exemple:

```
touch ex01/main.c
touch ex02/README.md
```

Si vous avez déjà des fichiers pour vos exercices, vous pouvez les déplacer dans les répertoires appropriés à l'intérieur du répertoire cloné:

```
mv /chemin/vers/vos/fichiers/* ex01/
mv /chemin/vers/vos/autres/fichiers/* ex02/
```

3. Ajouter des Fichiers à la Staging Area

Ajouter un fichier spécifique à la staging area:

```
git add ex01/main.c
```

Explication:

`git add` : Cette commande ajoute des fichiers à la staging area. La staging area est une zone où vous préparez les fichiers avant de les committer. Elle permet de regrouper les modifications que vous souhaitez committer ensemble.

Ajouter tous les fichiers dans un dossier à la staging area:

```
git add ex02/
```

Ajouter tous les fichiers modifiés à la staging area:

```
git add .
```

Explication:

`git add .` : Cette commande ajoute tous les fichiers modifiés dans le répertoire courant et ses sous-répertoires à la staging area.

4. Vérifier l'État du Dépôt

Vérifier les fichiers ajoutés à la staging area:

```
git status
```

Explication:

`git status` : Cette commande est utilisée après `git add` pour vérifier quels fichiers ont été ajoutés à la staging area et sont prêts à être committés.

5. Valider les Modifications

Committer les fichiers ajoutés avec un message descriptif:

```
git commit -m "Ajout des exercices ex01 et ex02"
```

Explication:

`git commit` : Cette commande enregistre les modifications dans l'historique du dépôt avec un message descriptif. Le texte après `-m` est le commit message, qui doit décrire les modifications apportées.

6. Pousser les Modifications vers le Dépôt Distant

Pousser les modifications sur le dépôt distant:

```
git push origin main
```

Explication:

`git push` : Cette commande envoie les commits locaux au dépôt distant sur GitHub. `origin` est le nom du dépôt distant par défaut, et `main` est la branche sur laquelle vous poussez les modifications.

Exemple Complet

Voici un exemple complet en suivant les étapes ci-dessus:

```
git clone https://votre_lien_git.git
cd nom_du_dépôt
mkdir ex01 ex02
touch ex01/main.c ex02/README.md
git add ex01/main.c ex02/
git status
git commit -m "Ajout des exercices ex01 et ex02"
git push origin main
```