

Commandes Git - Guide Personnalisé pour Tarak

Voici un guide des commandes Git avec des explications étape par étape pour Tarak. Ce guide vous permettra de suivre les bonnes pratiques pour versionner votre code avec Git.

Étape 1 : Configurer Git

Avant de commencer, configurez Git avec votre nom et votre email.

Commandes :

```
git config --global user.name "Tarak"
```

```
git config --global user.email "votreemail@example.com"
```

Étape 2 : Initialiser un dépôt Git

Pour commencer à versionner votre projet, vous devez initialiser un dépôt Git dans votre répertoire.

Commande :

```
cd /chemin/vers/votre/projet
```

```
git init
```

Étape 3 : Ajouter des fichiers au dépôt

Après avoir créé des fichiers ou apporté des modifications, vous devez les ajouter au suivi de Git.

Commandes :

```
git add fichier.txt
```

OU pour ajouter tous les fichiers :

```
git add .
```

Étape 4 : Créer un commit

Une fois les fichiers ajoutés, créez un commit pour enregistrer les modifications dans l'historique de Git.

Commande :

```
git commit -m "Description du commit"
```

Étape 5 : Créer un dépôt GitHub

Allez sur GitHub et créez un nouveau dépôt. Copiez ensuite l'URL du dépôt.

Étape 6 : Lier le dépôt local à GitHub

Pour connecter votre dépôt local à GitHub, utilisez cette commande (remplacez par l'URL de votre dépôt) :

Commande :

```
git remote add origin https://github.com/toncompte/tondepot.git
```

Étape 7 : Pousser le code vers GitHub

Une fois que tout est configuré, vous pouvez pousser (envoyer) votre code vers GitHub.

Commande :

```
git push -u origin master
```

Étape 8 : Vérifier l'état des modifications

Pour savoir si vous avez des modifications non suivies ou non commit :

Commande :

```
git status
```

Étape 9 : Travailler avec des branches

Les branches sont importantes pour travailler sur des fonctionnalités ou des correctifs sans impacter la branche principale.

Créer une nouvelle branche :

```
git checkout -b ma-branche
```

Changer de branche :

```
git checkout master
```

Fusionner une branche dans master :

```
git merge ma-branche
```

Étape 10 : Annuler des modifications

Si vous avez besoin d'annuler des modifications avant de les committer, vous pouvez utiliser :

Commande :

```
git checkout -- fichier.txt
```