

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

1. REFERENCE DE LA CORBEILLE

Version	Date	Concepteurs	Relecteurs	Commentaire
1.0	27/07/2024	DN		Première version

Objectif

L'objectif de ce guide est de vous apprendre à utiliser Git et GitHub pour gérer le code source d'un projet, suivre les modifications, collaborer avec d'autres développeurs, et maintenir un historique propre et organisé.

Étape 1 : Installation de Git

Sous Windows :

Téléchargez et installez [Git pour Windows](#). Une fois installé, Git est disponible à partir de l'invite de commandes ou PowerShell.

Sous MacOS :

- Si Homebrew n'est pas déjà installé, suivez les instructions sur <https://brew.sh> pour l'installer.
- Ouvrez un terminal et exécutez la commande : **brew install git**.

Où

- Téléchargement depuis le site Web : - Rendez-vous sur <https://git-scm.com/download/mac> et téléchargez le programme d'installation.
- Exécutez le programme d'installation téléchargé et suivez les instructions.

Sous Linux :

- Utilisation d'un gestionnaire de paquets (Ubuntu/Debian) : - Ouvrez un terminal et exécutez la commande : **sudo apt update**.
- Ensuite, installez Git avec la commande : **sudo apt install git**.
- Utilisation d'un gestionnaire de paquets (Fedora) : - Ouvrez un terminal et exécutez la commande : **sudo dnf install git**.
- Utilisation d'un gestionnaire de paquets (openSUSE) : - Ouvrez un terminal et exécutez la commande : **sudo zypper install git**.

1.1 Vérifier si Git est installé

- **Question** : Quelle commande utilisez-vous pour vérifier si Git est installé sur votre machine ?
- **Attendu** : Vous devriez voir la version de Git installée. Si Git n'est pas installé, passez à l'étape suivante.

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

Étape 2 : Configuration de Git

2.1 Configurer votre nom d'utilisateur et votre adresse e-mail

- **Question** : Comment configurez-vous votre nom d'utilisateur Git ?
- **Question** : Pourquoi est-il important de configurer votre nom d'utilisateur et votre adresse e-mail ?

Étape 3 : Initialisation d'un dépôt Git

3.1 Initialiser un nouveau dépôt Git sur votre machine

- **Question** : Quelle commande utilisez-vous pour initialiser un nouveau dépôt Git dans un répertoire existant ?
- **Question** : Que fait cette commande ?

Étape 4 : Suivi des changements

Créez un fichier index.html avec le code suivant :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

</head>

<body>

  <p>Hello World Guys</p>

</body>

</html>
```

4.1 Ajouter des fichiers à la zone de staging

- **Question** : Comment ajoutez-vous un fichier spécifique à la zone de staging ?

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

- **Question** : Quelle commande utilisez-vous pour ajouter tous les fichiers modifiés à la zone de staging ?
- **Question** : Comment vérifiez-vous l'état des fichiers dans le dépôt ?

Étape 5 : Validation des changements

5.1 Valider les changements ajoutés à la zone de staging

- **Question** : Quelle commande utilisez-vous pour valider les changements ajoutés à la zone de staging ?
- **Question** : Comment incluez-vous un message de commit clair et descriptif ?

Étape 6 : Historique des commits

6.1 Afficher l'historique des commits

- **Question** : Quelle commande utilisez-vous pour afficher l'historique des commits ?
- **Question** : Comment affichez-vous les détails d'un commit spécifique ?

Étape 7 : Gestion des branches

7.1 Créer et basculer entre les branches

- **Question** : Comment créez-vous une nouvelle branche dans Git ?
- **Question** : Quelle commande utilisez-vous pour basculer entre les branches ?
- **Question** : Comment fusionnez-vous une branche dans la branche principale ?

Étape 8 : Rejet des modifications

8.1 Annuler les modifications non validées

- **Question** : Comment annulez-vous les modifications non validées dans un fichier ?
- **Question** : Quelle commande utilisez-vous pour réinitialiser un commit spécifique ?

Étape 9 : Connexion à GitHub

9.1 Créer un nouveau dépôt sur GitHub

- **Question** : Comment créez-vous un nouveau dépôt sur GitHub ? Créez un nouveau dépôt du nom de : **Hello-World-Git**
- **Question** : Quelle commande utilisez-vous pour ajouter un dépôt distant GitHub à votre dépôt local ?

Étape 10 : Pousser les changements vers GitHub

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

10.1 Pousser vos commits vers le dépôt distant sur GitHub

- **Question** : Quelle commande utilisez-vous pour pousser vos commits vers le dépôt distant sur GitHub ?
- **Question** : Comment configurez-vous Git pour pousser automatiquement à une branche spécifique sur GitHub ?

Étape 11 : Collaboration avec d'autres développeurs

11.1 Cloner un dépôt existant depuis GitHub

- **Question** : Comment clonez-vous un dépôt existant depuis GitHub ?

11.2 Récupérer les modifications depuis le dépôt distant

- **Question** : Quelle commande utilisez-vous pour récupérer les modifications depuis le dépôt distant ?

11.3 Gestion des conflits de fusion

- **Question** : Comment gérez-vous les conflits de fusion lors de la collaboration avec d'autres développeurs ?

Étape 12 : Utilisation des pull requests

12.1 Créer une pull request sur GitHub

- **Question** : Comment créez-vous une pull request sur GitHub ?
- **Question** : Quelle est l'importance des pull requests dans le flux de travail collaboratif ?

Étape 13 : Bonnes pratiques de Git et GitHub

13.1 Écrire des messages de commit clairs et significatifs

- **Question** : Quelles sont les bonnes pratiques pour écrire des messages de commit clairs et significatifs ?

13.2 Organiser votre dépôt

- **Question** : Comment organisez-vous votre dépôt pour une meilleure gestion du code source ?

Étape 14 : Introduction à Git Flow

14.1 Comprendre Git Flow

- **Question** : Qu'est-ce que Git Flow et pourquoi est-il utilisé dans la gestion de projet ?

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

Étape 15 : Installation de Git Flow

15.1 Installer Git Flow

- **Question** : Quelle commande utilisez-vous pour installer Git Flow sur votre système ?

Étape 16 : Initialisation de Git Flow

16.1 Initialiser Git Flow dans un dépôt Git existant

- **Question** : Quelle commande utilisez-vous pour initialiser Git Flow dans un dépôt Git existant ?
- **Question** : Que se passe-t-il lorsque vous initialisez Git Flow ? Quels sont les types de branches créés par défaut ?

Étape 17 : Utilisation des branches Git Flow

17.1 Travailler sur une nouvelle fonctionnalité

- **Question** : Quelle commande utilisez-vous pour créer et basculer sur une nouvelle branche de fonctionnalité ?
- **Question** : Comment terminez-vous une branche de fonctionnalité une fois que le développement est terminé ?

17.2 Préparer une release

- **Question** : Quelle commande utilisez-vous pour commencer une nouvelle branche de release ?
- **Question** : Que faites-vous après avoir terminé le développement sur une branche de release ?

17.3 Gestion des hotfixes

- **Question** : Quelle commande utilisez-vous pour commencer une nouvelle branche de hotfix ?
- **Question** : Comment terminez-vous une branche de hotfix ?

Étape 18 : Collaboration avec Git Flow

18.1 Partager des branches Git Flow

- **Question** : Comment poussez-vous une branche de fonctionnalité vers le dépôt distant pour que d'autres développeurs puissent collaborer ?

18.2 Revue de code et pull requests

- **Question** : Comment utilisez-vous les pull requests en combinaison avec Git Flow pour faciliter les revues de code ?

Étape 19 : Bonnes pratiques avec Git Flow

19.1 Adopter des conventions de nommage

Corbeille : 3 – Gestion de versions

GUIDE TUTEUR

- **Question** : Quelles sont les conventions de nommage pour les branches de fonctionnalité, de release, et de hotfix dans Git Flow ?

19.2 Maintenir un historique de commits propre

Question : Comment utilisez-vous les commandes rebase et squash pour maintenir un historique de