

 Review the assignment due date

Guide d'Utilisation pour le Projet [Sixteen-Soldiers]

Bienvenue dans le projet **Sixteen-Soldiers** ! Ce document vous guidera étape par étape pour :

- [Créer un compte GitHub](#)
 - [Installer Git sur votre machine](#)
 - [Accéder à la GitHub Classroom et rejoindre une équipe](#)
 - [Cloner le projet](#)
 - [Travailler sur le projet et pousser vos modifications](#)
 - [Découvrir des ressources utiles](#)
-

1. Créer un Compte GitHub

Si vous n'avez pas encore de compte GitHub, suivez ces étapes simples :

1. Allez sur github.com.
2. Cliquez sur **Sign up** pour créer un compte.
3. Remplissez les informations demandées (email, mot de passe, etc.).
4. Confirmez votre email en suivant le lien envoyé dans votre boîte de réception.

Conseils pour la sécurité :

- Choisissez un mot de passe solide.
 - Activez l'authentification à deux facteurs pour renforcer la sécurité de votre compte.
-

2. Installer Git

a) Windows

1. Téléchargez le programme d'installation ici : git-scm.com.
2. Suivez les instructions par défaut pour l'installation.

b) macOS

1. Installez Git avec **Homebrew** (si installé) :

```
brew install git
```

2. Sinon, téléchargez-le depuis git-scm.com.

c) Linux

1. Installez Git via votre gestionnaire de paquets (par exemple, pour Ubuntu) :

```
sudo apt update  
sudo apt install git
```

Vérification de l'installation :

Dans votre terminal, exécutez :

```
git --version
```

Configuration Globale de Git

Voici les commandes essentielles pour configurer Git après installation :

1. **Configurer votre nom d'utilisateur Git :**

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

2. **Configurer votre email Git :**

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

3. **Configurer les couleurs pour un affichage plus clair :**

```
git config --global color.ui auto
```

4. **Vérifier vos configurations globales :**

```
git config --list
```

Avec ces configurations, vous serez prêt à utiliser Git efficacement ! 🚀

👥 3. Rejoindre la GitHub Classroom

1. Cliquez sur le lien d'invitation reçu par email pour accéder à la **GitHub Classroom**.
2. Connectez-vous avec votre compte GitHub.
3. Suivez les étapes pour :
 - **Créer une équipe** (si vous êtes capitaine).
 - **Rejoindre une équipe existante** (si vous faites partie d'un groupe).

🔑 Points Clés :

- **Capitaines d'équipe :**

- Créez l'équipe via le lien fourni dans la Classroom.
- Donnez un nom clair et cohérent à votre équipe, en suivant les instructions fournies (par ex. "Team Alpha").
- Assurez-vous que le nom de l'équipe correspond exactement à celui défini dans les consignes.

- **Membres d'équipe :**

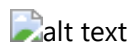
- Une fois l'équipe créée, rejoignez-la via la liste des équipes disponibles dans la GitHub Classroom.

🔔 **Rappel :** Tous les membres doivent être assignés à une équipe via la plateforme pour que leurs contributions soient prises en compte.

💻 4. Cloner le Projet

Après avoir rejoint une équipe dans la GitHub Classroom :

1. Vous serez redirigé vers une page contenant un lien vers votre dépôt d'équipe dont voici un aperçu :



2. Cliquez sur ce lien pour accéder au dépôt GitHub associé.
3. Pour cloner le projet sur votre machine :

- Copiez l'URL du dépôt via le bouton **Code** > **HTTPS**.
- Dans votre terminal, exécutez :

```
git clone [URL du dépôt]
```

Remplacez [URL du dépôt] par l'URL que vous avez copiée.

🔧 5. Travailler sur le Projet

a) Modifier des fichiers

- Ajoutez ou modifiez des fichiers localement avec votre éditeur de code préféré (VS Code, PyCharm, etc.).

b) Commandes Git essentielles

- **Voir les fichiers modifiés :**

```
git status
```

- **Ajouter un fichier spécifique :**

```
git add nom_du_fichier.txt
```

- **Ajouter tous les fichiers modifiés :**

```
git add .
```

c) Enregistrer vos modifications (commit)

Une fois les fichiers ajoutés, enregistrez vos modifications avec un message explicatif :

```
git commit -m "Description des modifications"
```

d) Envoyer vos modifications sur GitHub (push)

1. Si vous travaillez sur la branche principale :

```
git push
```

2. Si vous travaillez sur une branche spécifique :

```
git push origin nom_de_votre_branche
```

⚠ **Important :** Poussez vos modifications régulièrement pour éviter tout problème et permettre leur prise en compte.

En effet, nous ne pourrons récupérer votre code que si vous faites un "push" de vos modifications. Toute équipe n'ayant pas poussé son code via les commandes sus-mentionnées ne verra d'office pas son code pris en compte et donc sera automatiquement recalée.

🌟 6. Quelques Ressources Utiles

- **Documentation GitHub** : <https://docs.github.com>
 - **Formation interactive GitHub** : [GitHub Learning Lab](#)
 - **Comprendre le flux GitHub** : [Introduction to GitHub Flow](#)
 - **Vidéos éducatives Git/GitHub** : [GitHub sur YouTube](#)
-

Avec ce guide, vous avez tout ce qu'il faut pour participer efficacement à la compétition **Sixteen-Soldiers**. Si vous avez des questions ou des problèmes, n'hésitez pas à demander de l'aide dans le canal whatsapp dédié.

Bon codage et amusez-vous bien ! 🍷

KYFAX ✨