

Exercice git

Scénario: Vous êtes une équipe de 3 développeurs et devez créer un petit script Python qui affiche une blague aléatoire en utilisant l'API de `pyjokes`. Votre projet doit être bien structuré et suivre les bonnes pratiques Git.

Étape 1 : Initialisation du Projet (1er membre de l'équipe)

- Un membre de l'équipe (Développeur A) crée un dépôt GitHub appelé **projet-blagues**.
- Il initialise Git **localement**
- Il initialise un projet Python et un environnement virtuel
- Il crée le fichier `requirements.txt` avec la dépendance suivante : `pyjokes`
- Il installe les dépendances **dans le venv**
- Il crée un fichier `main.py` contenant le script suivant:

```
import pyjokes

def tell_joke():
    print(pyjokes.get_joke())

if __name__ == "__main__":
    tell_joke()
```

- Il crée un fichier `.gitignore` pour ignorer l'environnement virtuel
- Il ajoute un fichier `README.md` avec le titre du projet et pousse le projet sur GitHub
- Les deux autres membres (DÉVELOPPEURS B et C) **clonent** le projet, créent et activent l'environnement virtuel
- Ils installent les dépendances depuis le fichier `requirements.txt`

Étape 2 : Création des branches individuelles

- Le deuxième développeur crée une nouvelle branche pour ajouter une fonctionnalité (ex: choix de la langue)
- Modifier `main.py` pour permettre de choisir la langue :

```
import pyjokes

def tell_joke(lang="en"):
    print(pyjokes.get_joke(language=lang))

if __name__ == "__main__":
    lang = input("Choisissez une langue (en, de, es, it, gl): ")
    tell_joke(lang)
```

- Le développeur B ajoute son fichier et fait un commit puis pousse sa branche sur github
- Le développeur C pull les changements, crée une nouvelle branche et modifie à son tour le code pour permettre de choisir le type de blague:

```
import pyjokes

def tell_joke(lang="en"):
    print(pyjokes.get_joke(language=lang))

if __name__ == "__main__":
    lang = input("Choisissez une langue (en, de, es, it, gl): ")
    joke_type = input("Choisissez une catégorie de blague (neutral, chuck, all): ")
    tell_joke(lang, joke_type)
```

- Le développeur C ajoute son fichier et fait un commit puis pousse sa branche sur github

Étape 3 : Fusionner les branches via Pull Requests

- Chaque membre ouvre une **Pull Request (PR)** sur GitHub pour fusionner sa branche dans `main`.
- Les autres membres doivent **relire et approuver** la PR.
- Une fois validée, la PR est fusionnée

Étape 5 : Gestion d'un conflit volontaire

- Deux développeurs modifient `main.py` (les mêmes lignes) en même temps sur des branches séparées.
- Ils poussent leurs modifications et essaient de fusionner → Un **conflit Git** apparaît !
- Ils doivent résoudre ce conflit **manuellement** avant de finaliser la fusion.

Étape 6 : Nettoyage et bonnes pratiques

- Chaque membre met à jour sa branche locale

git checkout main

git pull origin main

- Supprimer les branches fusionnées