

TP Complet : Branches & Pull Requests

Création du projet local

A. Créer un dossier de travail

Créer un dossier : **TP_Git_Branches**

```
mkdir TP_Git_Branches  
cd TP_Git_Branches
```

B. Initialiser Git

```
git init
```

C. Créer un fichier

Créer un fichier **app.txt** :

Version initiale de l'application.

D. Ajouter puis commit

```
git add app.txt  
git commit -m "Initial commit : ajout de app.txt"
```

Créer un dépôt GitHub & connecter

1. Aller sur GitHub → New Repository
2. Nommer : **TP_Git_Branches**
3. Ne pas ajouter de README

Puis connecter :

```
git remote add origin https://github.com/votre_nom/TP_Git_Branches.git  
git branch -M main  
git push -u origin main
```

Travail avec les branches

A. Créer une nouvelle branche

Créer une branche nommée *feature1* :

```
git branch feature1
```

Ou directement changer de branche :

```
git checkout -b feature1
```

B. Modifier le fichier

Dans `app.txt`, ajouter :

Ajout de la fonctionnalité 1.

Ensuite commit :

```
git add app.txt
git commit -m "Ajout fonctionnalité 1"
```

C. Envoyer la branche sur GitHub

```
git push -u origin feature1
```

Création d'une Pull Request (PR)

1. Aller sur GitHub → votre dépôt.
2. GitHub détectera la branche *feature1* et proposera **Compare & Pull Request**.
3. Cliquer → ajouter un titre :
Ajout de la fonctionnalité 1
4. Ajouter une description (ex : "Ajout d'une nouvelle section dans app.txt").
5. Cliquer **Create Pull Request**.

À ce stade, la fonctionnalité attend la validation d'un responsable.

Validation de la Pull Request

En tant qu'enseignant ou collègue, vous pouvez :

- Consulter le code modifié.
- Ajouter un commentaire.
- Approuver la PR.
- Fusionner en cliquant sur **Merge Pull Request**.

Après fusion, supprimer la branche (option proposée).

Exercice avec conflit volontaire

Objectif : comprendre les conflits et les résoudre.

Étape 1 : Étudiant A crée une branche

```
git checkout -b update_A
```

Ajouter dans `app.txt` :

Ligne modifiée par A.

Commit & push :

```
git add .
git commit -m "Modification par A"
git push -u origin update_A
```

Créer une Pull Request mais **ne pas la merger**.

Étape 2 : Étudiant B modifie la même ligne

```
git checkout -b update_B
```

Dans le même fichier `app.txt` :

Ligne modifiée par B.

Commit & push :

```
git add .
git commit -m "Modification par B"
git push -u origin update_B
```

Créer aussi une Pull Request.

Étape 3 : Fusionner la PR de A

Le professeur merge la PR de A en premier.

Étape 4 : Essayer de fusionner la PR de B

GitHub affichera :

This branch has conflicts that must be resolved.

Le conflit se présente dans `app.txt`.

Étape 5 : Résolution du conflit

Étudiant B récupère les mises à jour :

```
git checkout update_B
git pull origin main
```

Git l'avertit du conflit.

Ouvrir `app.txt`, vous verrez :

```
<<<<<<< HEAD
Ligne modifiée par B.
=====
Ligne modifiée par A.
>>>>>>> main
```

Résoudre manuellement, par exemple :

Ligne modifiée par A et B.

Puis commit :

```
git add app.txt
```

```
git commit -m "Résolution du conflit"  
git push
```

Retourner sur GitHub : la PR peut maintenant être fusionnée.

Suppression des branches après fusion

```
git branch -d feature1  
git push origin --delete feature1
```