

# TP Git — Merge et Résolution de Conflits

> Format court et progressif, avec énoncés et vérifications attendues.

---

## Pré-requis

- Git installé (`git --version`)
- Un terminal (Bash, PowerShell, Git Bash, etc.)

---

## Partie 1 — Préparation du dépôt

### 1) Créer un nouveau repository Git

```
mkdir tp-merge && cd tp-merge
git init
```

### 2) Ajouter un fichier initial et le committer

```
echo "Ligne 1 - version initiale" > index.txt
git add index.txt
git commit -m "Initial commit avec index.txt"
```

---

## Partie 2 — Création de branches et modifications

### 3) Créer une branche `feature-a` et modifier le fichier

```
git checkout -b feature-a
echo "Ligne 2 - ajoutée dans feature-a" >> index.txt
git add index.txt
git commit -m "Ajout ligne 2 dans feature-a"
```

### 4) Revenir sur la branche principale et modifier aussi le fichier

```
git checkout main # ou master selon la version
echo "Ligne 2 - ajoutée dans main" >> index.txt
git add index.txt
git commit -m "Ajout ligne 2 dans main"
```

---

## Partie 3 — Fusion et Conflit

### 5) Fusionner la branche `feature-a` dans `main`

```
git merge feature-a
```

■ Git va signaler un **conflit** car `index.txt` a été modifié différemment dans les deux branches.

### 6) Vérifier l'état

```
git status
```

\_Attendu : \_ `both modified: index.txt`.

---

## Partie 4 — Résolution du conflit

### 7) Ouvrir le fichier `index.txt`

Le fichier contiendra des marqueurs :

Ligne 1 - version initiale

<<<<<< HEAD

Ligne 2 - ajoutée dans main

=====

Ligne 2 - ajoutée dans feature-a

>>>>>> feature-a

### 8) Résoudre manuellement

Décidez du contenu final, par exemple :

Ligne 1 - version initiale

Ligne 2 - ajoutée dans main + feature-a

### 9) Ajouter et valider la résolution

git add index.txt

git commit -m "Fusion de feature-a dans main (résolution conflit)"

---

## Partie 5 — Vérification

### 10) Vérifier l'historique

git log --oneline --graph --all

\_Attendu : \_ un commit de merge reliant les deux branches.

### 11) Lister les branches

git branch

\_Attendu : \_ `main` et `feature-a`.

---

## Bonus (facultatif)

- Créer un merge **\*\*sans conflit\*\*** : modifier deux fichiers différents dans deux branches puis merger.

- Forcer un **\*\*merge non fast-forward\*\*** :

git merge --no-ff feature-a -m "Merge sans fast-forward"

---

## Récapitulatif des commandes clés

git checkout -b <branch>

```
git merge <branch>  
git status  
git add <fichier>  
git commit -m "..."  
git log --oneline --graph --all
```