# Atelier Git avec Bitbucket - Plan de cours et exercices corrigés

## Objectifs

- Comprendre le fonctionnement de Git (local et distant)  
- Travailler avec Bitbucket pour héberger et gérer ses dépôts  
- Apprendre à collaborer via des branches et pull requests

## Prérequis

- Git installé sur votre machine  
- Compte Bitbucket actif  
- Un éditeur de code (ex : VS Code)

## Partie 1 : Introduction à Git (30 min)

Objectifs : Comprendre le rôle de Git et exécuter des commandes de base en local.

* Exercice 1 : Créer un dépôt local

mkdir projet-demo  
cd projet-demo  
git init  
echo "Hello Git" > README.md  
git status  
git add README.md  
git commit -m "Premier commit"  
git log

Correction : Vérifier que le commit est enregistré avec `git log`.

## Partie 2 : Branches et fusion (30 min)

* Exercice 2 : Créer une branche et la fusionner

git checkout -b feature-hello  
echo "Hello depuis feature" >> README.md  
git commit -am "Ajout message dans feature"  
git checkout main  
git merge feature-hello

Correction : Vérifier que le message ajouté est bien visible dans README.md.

## Partie 3 : Introduction à Bitbucket (20 min)

* Exercice 3 : Créer un dépôt distant et le relier

git remote add origin https://bitbucket.org/<votre-utilisateur>/projet-demo.git  
git push -u origin main

Correction : Vérifier la présence des fichiers sur Bitbucket.

## Partie 4 : Collaboration avec Bitbucket (1h)

* Exercice 4 : Pull Request avec revue de code

git checkout -b feature-footer  
echo "<footer>Site Web</footer>" > footer.html  
git add footer.html  
git commit -m "Ajout du footer"  
git push -u origin feature-footer

Sur Bitbucket : Créer une PR, faire une revue, puis fusionner.

## Partie 5 : Résolution de conflits (30 min)

* Exercice 5 : Simuler un conflit

# Sur main :  
echo "Version A" > conflit.txt  
git add .  
git commit -m "Version A"  
  
# Sur une branche :  
git checkout -b conflit-branch  
echo "Version B" > conflit.txt  
git add .  
git commit -m "Version B"  
  
# Fusion :  
git checkout main  
git merge conflit-branch

Correction : Résoudre le conflit manuellement, puis commit.

## Partie 6 : Bonnes pratiques (10 min)

- Une branche = une fonctionnalité  
- Des commits fréquents et explicites  
- Petites PR faciles à relire  
- Toujours faire un pull avant de commencer à coder