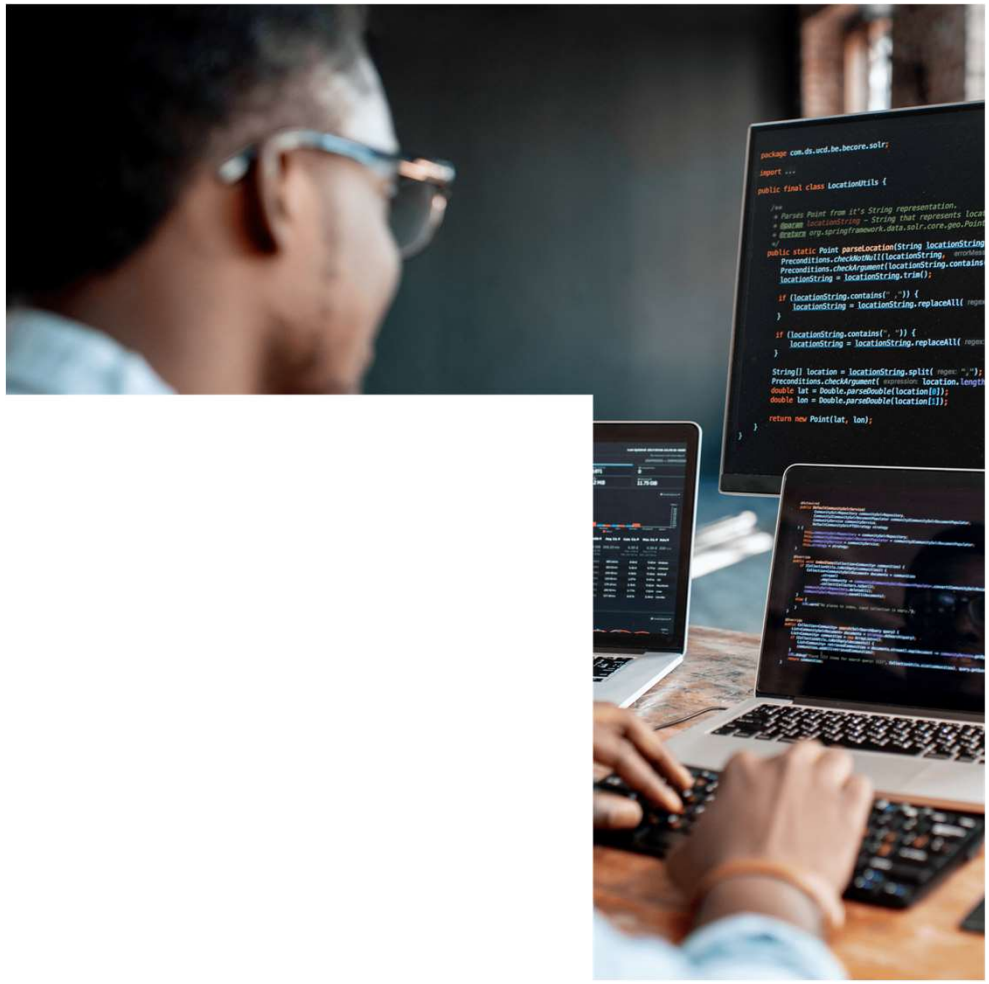
Pratique Git

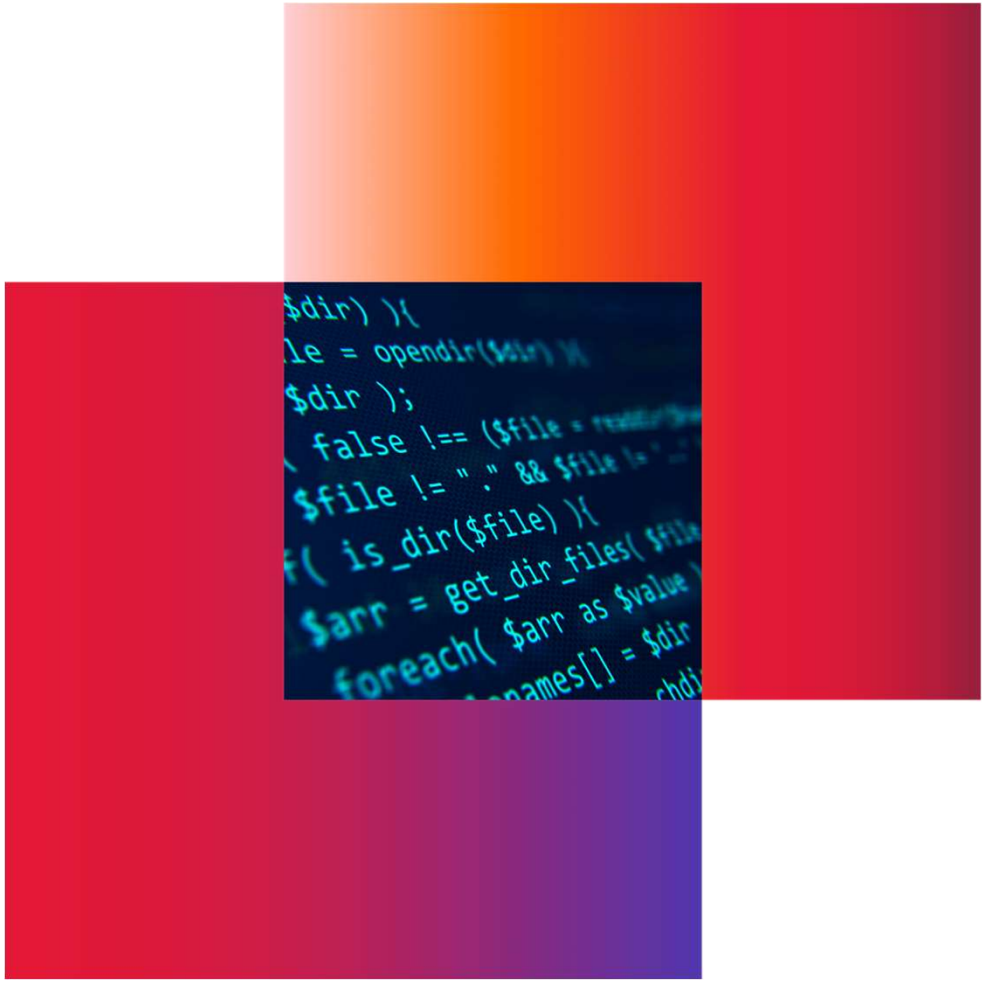
Développeur Web



Signes & Formations

Promo 2023-2024

Premiers pas



Récupérer le repo

1. Utilisez la commande **git clone** pour copier le repository à l'adresse indiqué par le formateur

2. Déplacez vous dans le nouveau répertoire créé puis utilisez la commande **git status** pour vérifier que vous êtes bien dans un répertoire GIT

3. Le formateur va ajouter un fichier dans le repo via **git add**, **git commit** et **git push**

4. Chaque personne peux ensuite effectuer un **git pull** et vérifier que le fichier est bien récupéré

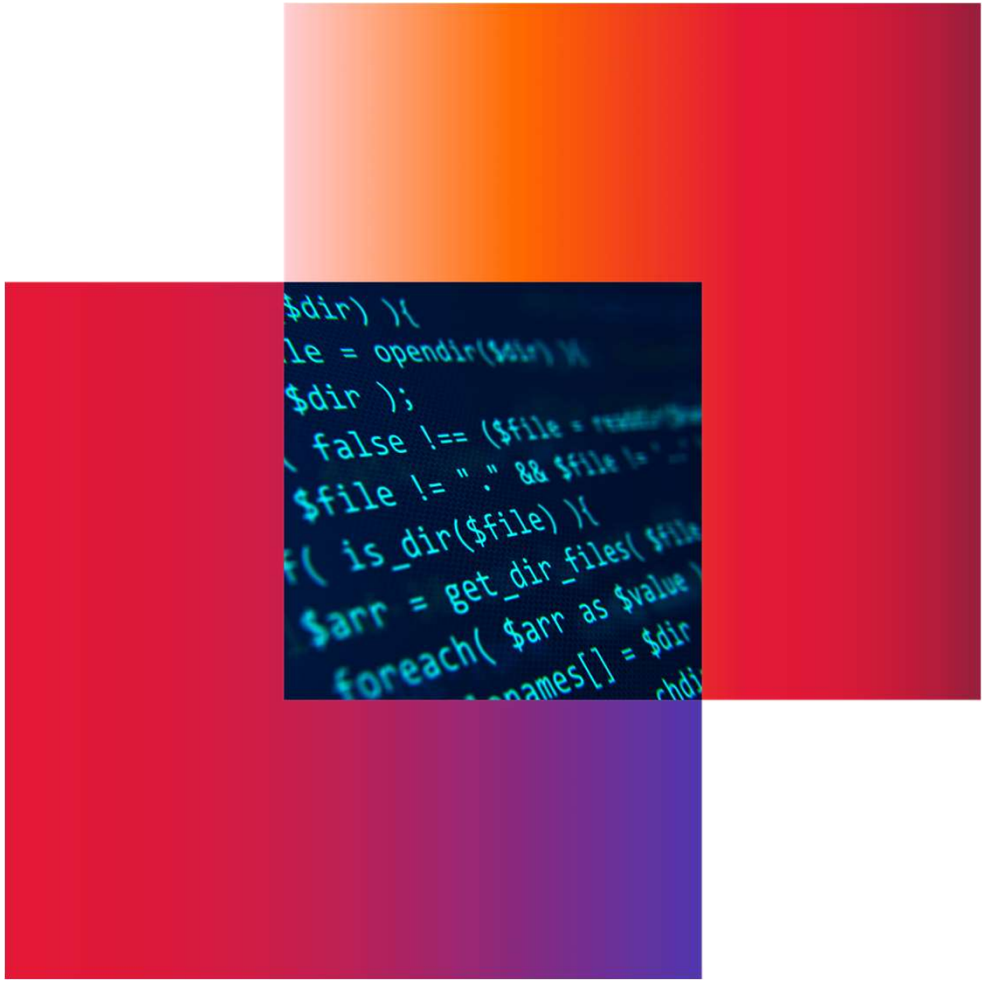
Ajouter un fichier

Chaque personne ajoute le fichier de son choix dans le repo local.

Ensuite chacun peux faire à son tour les étapes suivantes :

1. Indexer le fichier avec **git add <nom du fichier>**
2. Lancer **git status** après chaque commande pour voir l'évolution
3. Pour envoyer le fichier sur le repo local, avec **git commit –m « message »**
4. Enfin, lancer **git fetch** puis **git push** pour envoyer le fichier sur le repo distant
5. **Git log** permet de voir l'historique des modifications

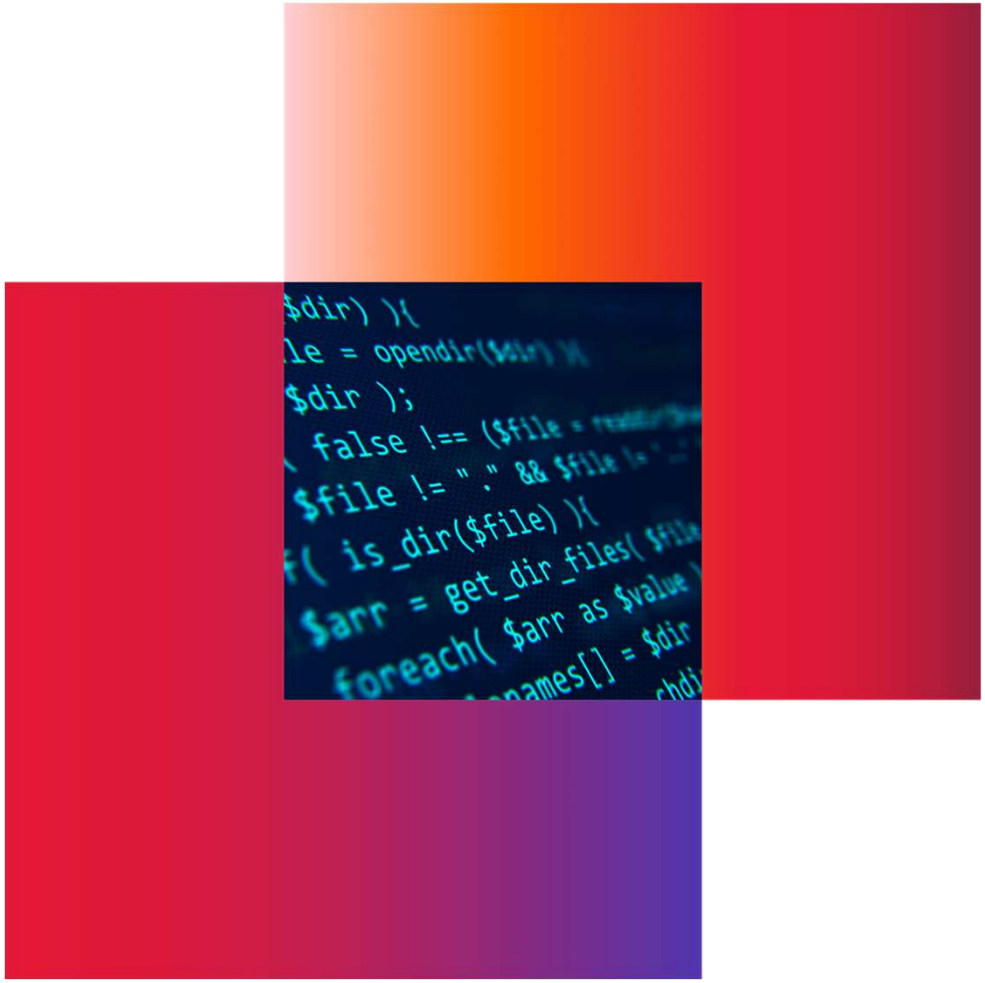
Naviguer sur les branches



Gestion des branches

1. Créez un nouveau dépôt dans un autre dossier
2. Initialisez le dépôt avec la commande **git init**
3. Créez un fichier texte (exemple : **touch hello.txt**)
4. Ajoutez puis commitez le fichier
5. Créez une branche **dev** et placez vous dans celle-ci
6. Modifiez le fichier « hello.txt » puis commitez les modifications
7. Revenez sur la branche **master**
8. Effectuez un merge de la branche **dev** sur la branche **master**
9. Vérifiez le contenu de « hello.txt »

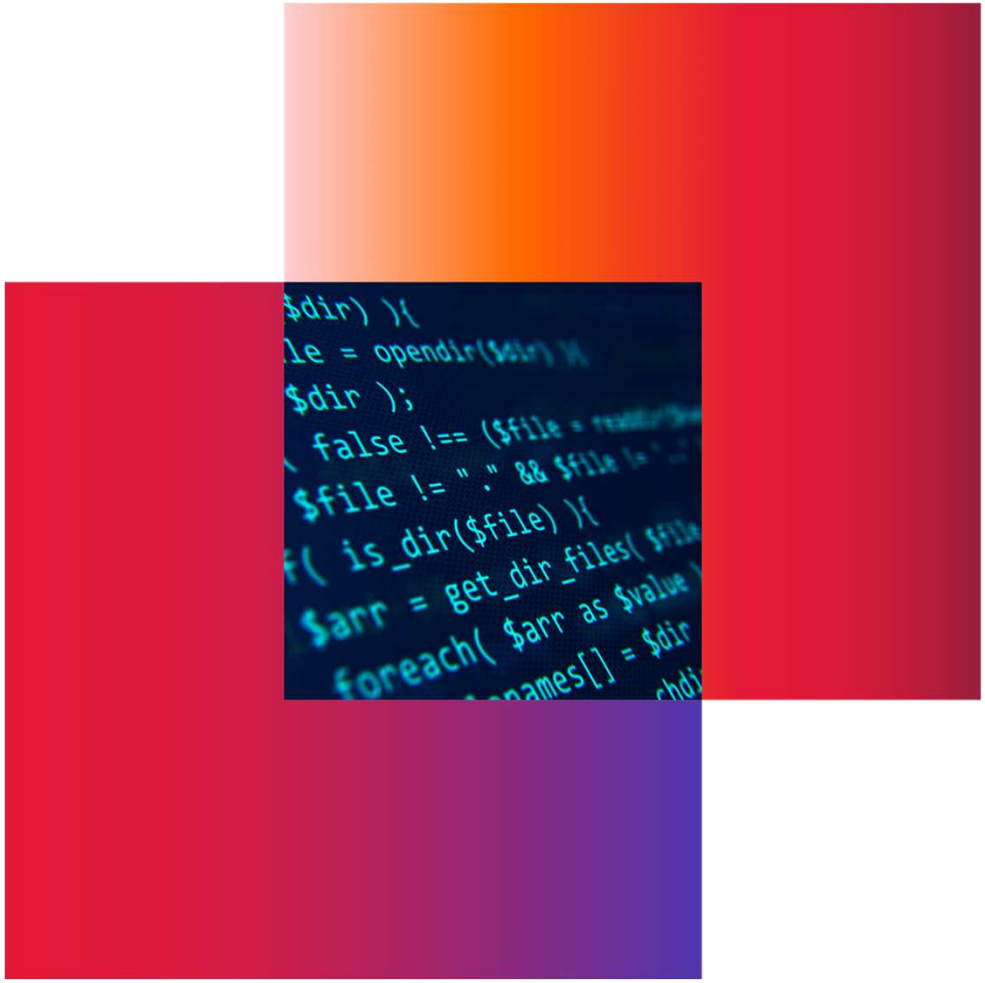
Gérer les conflits



Gestion de conflits

1. Replacez vous sur la branche **dev**
2. Créez une branche **feature1** et placez vous dans celle ci
3. Modifiez le fichier « hello.txt » puis commitez les modifications
4. Revenez sur la branche **dev**
5. Créez une branche **feature2** et placez vous dans celle ci
6. Modifiez à nouveau le fichier « hello.txt », commitez les modifications puis revenez sur la branche **dev**
7. Effectuez un merge de la branche **feature1** sur la branche **dev**
8. Vérifiez le contenu de « hello.txt »
9. Effectuez un merge de la branche **feature2** sur la branche **dev**
10. Un conflit apparait !
11. Vérifiez le contenu du fichier « hello.txt »
12. Gérez le conflit en modifiant le fichier manuellement puis commitez

Exercices supplémentaires



Naviguer entre différents commits

1. Placez vous sur la branche **master**

2. Utilisez **git log** pour voir la liste des commits effectuez

3. Utilisez la commande **git checkout** pour revenir sur un précédent commit

4. Revenir au dernier commit effectué avec la commande **git checkout –** (ou **git checkout master** selon votre version de GIT)

Sauvegarder temporairement ses modifications

1. Placez vous sur la branche **feature1**
2. Effectuez n'importe quelles modifications sans les commiter
3. Utilisez la commande **git stash** pour sauvegarder temporairement ces modifications
4. Retournez sur la branche master puis retournez à nouveau sur la branche **feature1**
5. Vos modifications précédentes ne sont pas visibles…
6. Rétablissez vos modification en utilisant **git stash apply**
7. Vérifiez que les modifications ont bien étés appliquées

Renommer et supprimer une branche

1. Renommez la branche **feature1** avec l'option **–m** de la commande **git branch**

2. Vérifiez que la commande s'est bien exécutée en listant les branches

3. Supprimez les branches **feature1** et **feature2** grâce à l'option **–d** de la commande **git checkout**

4. Vérifiez que la suppression s'est bien exécutée en listant les branches

Rectifier un commit en cas d’erreur

1. Créez deux fichier « ok.txt » et « oubli.txt »
2. Indexez puis commitez le fichier « ok.txt »
3. Faites un **git status** pour vérifier l’état des fichiers : le fichier « oubli.txt » n’a pas été indexé
4. Indexez le fichier « oubli.txt »
5. Utilisez l’option **--amend** de la commande git commit pour rectifier le commit
6. Vérifiez que le commit a bien été rectifié avec la commande **git log**

Déplacer la base d’une branche

1. Créez une nouvelle branche **experiment**
2. Créez un fichier « a.txt » puis indexez et commitez la modification
3. Retournez sur la branche **master**
4. Créez un fichier « b.txt » puis indexez et commitez la modification
5. Lancez la commande **git rebase experiment**
6. L’ajout du fichier b.txt a du être placé à la suite de l’ajout du fichier a.txt. Vous pouvez vérifier cela avec la commande **git log**

Bon courage !