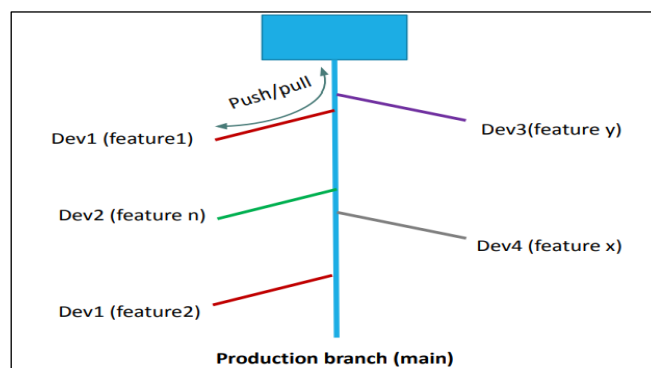


TP3 : Notion de branches

La notion de la branche en git permet de travailler simultanément avec plusieurs versions du code source dans le référentiel GitLocal. Par exemple, si un développeur voulait développer un feature, il le fait ressortir dans une branche à part. Ainsi, le feature sera codé et testé au niveau de la branche sans toucher à la branche principale. Après avoir assuré qu'il fonctionne bien, on pourra le fusionner dans la branche principale (main, master).

Le fait de créer une branche à part consiste à créer un pointeur virtuel pointant sur toutes les opérations effectuées au niveau de cette branche.



1. Tapez la commande « git branch », Quel est le résultat obtenu ?
2. Créez une nouvelle branche « Branche1 » avec la commande « git branch " nom-de-la-branche" ».
3. Vérifiez la création de la nouvelle branche.
4. Switchez vers la branche créée avec la commande « git checkout " nom-de-la-branche" ».
5. Vérifiez l'emplacement actuel du pointeur.
6. Switchez vers la branche créée, ajouter un fichier dedans (filebranch1.txt) par la suite commiter le, Switchez vers main et consultez tous les commits. Que remarquez-vous ?
7. Uploader la nouvelle branche dans le serveur distant avec la commande « git push origin Nom de branche » que remarquez-vous ?
8. Modifiez le contenu du fichier et consulter les modifications apportées avec la commande « git diff NomFichier »
9. Modifiez les deux derniers commits et vérifiez la différence entre ces deux derniers avec la commande « git diff "id de 1er commit" "id de 2eme commit" »
10. Consulter la différence entre les deux derniers commits et sur quels fichiers ils ont été effectués avec la commande « git diff "id de 1er commit" "id de 2eme commit" -- name oneline »

Supposons qu'on a 15 copies de code source chez les développeurs. Un parmi ces développeurs a créé une branche à part, et a codé sa partie avec plusieurs modifications en local, il a trouvé des bugs qu'il n'a pas pu les résoudre. Quelle est la solution !!

11. Créez un fichier, après modifiez le et faites un roolback avec la commande « `git checkout -- Nom de fichier` »(On peut retourner à l'état initial de tous les fichiers au niveau d'un dossier courant avec la commande « `git checkout --` »

On pourra créer et switcher dans une branche à la fois avec la commande « ***git checkout -b "nom-de-la-branche"*** ».

Pour renommer une branche, on utilise la commande « ***git branch -m " Ancien_Nom" " Nouveau_Nom"***».