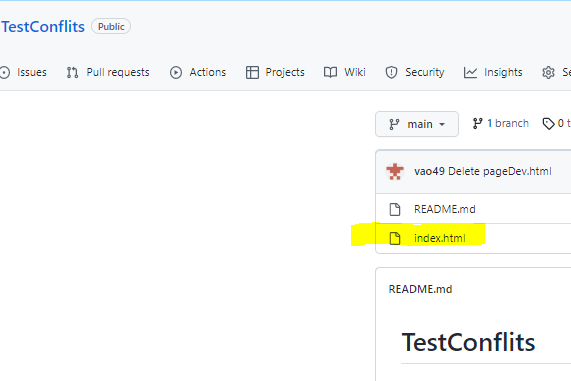
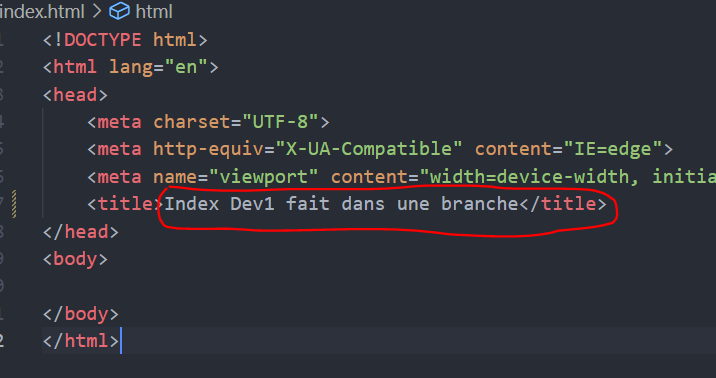
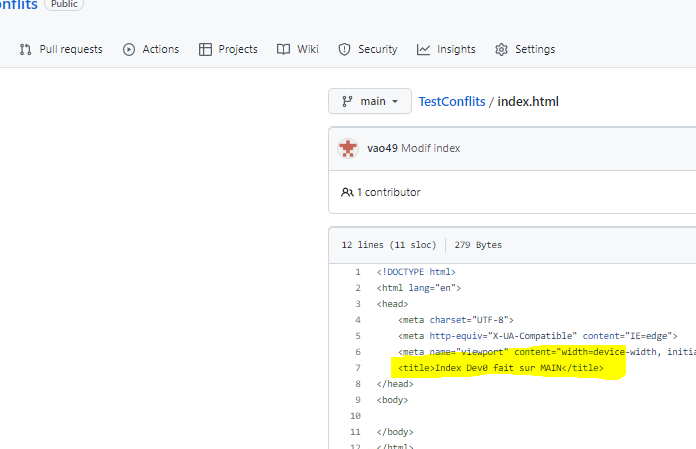
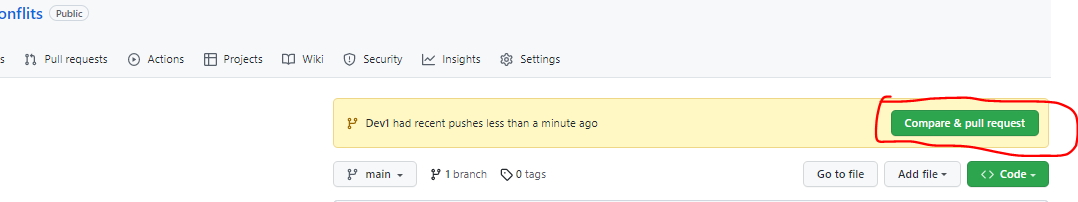
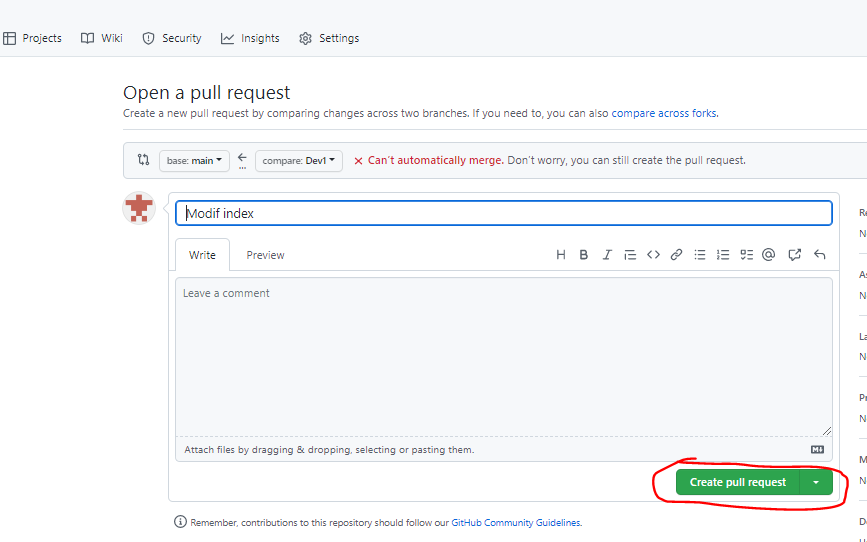
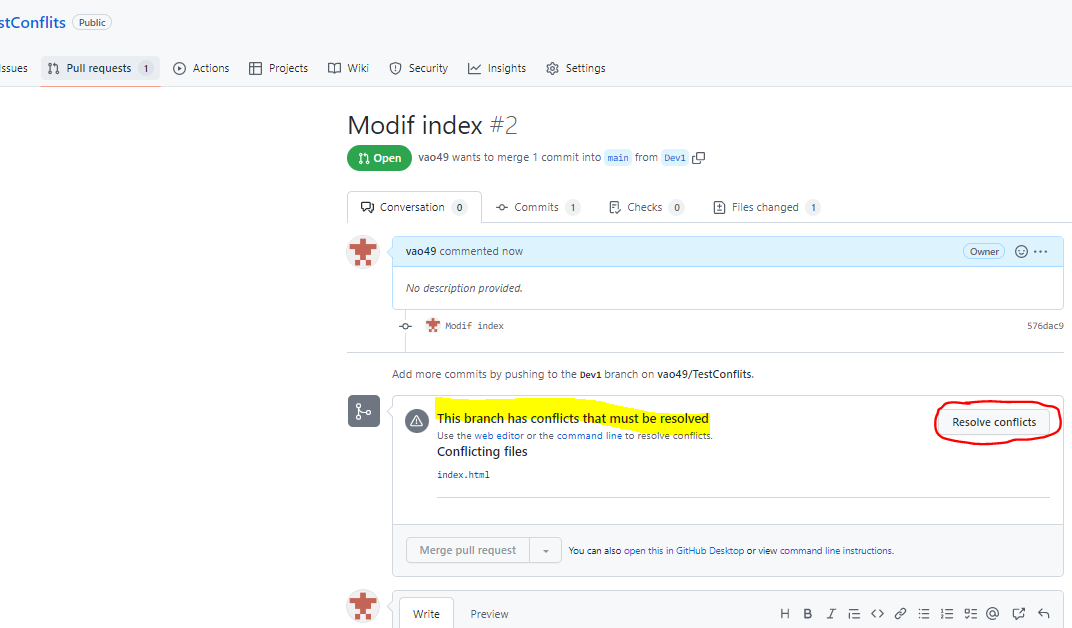
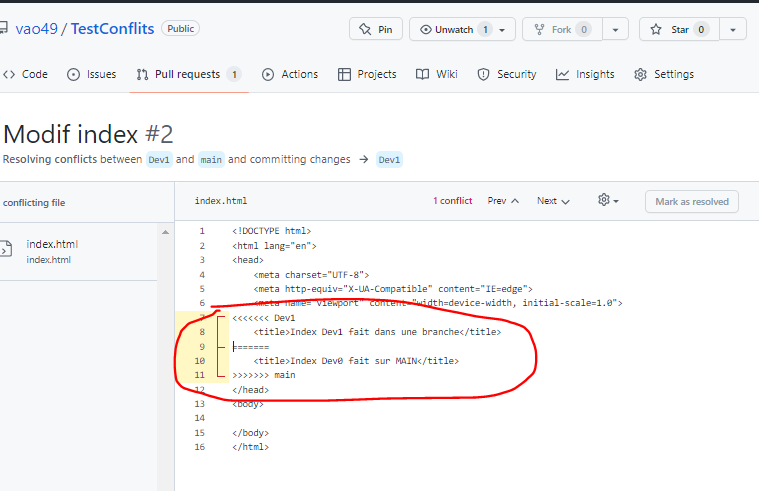
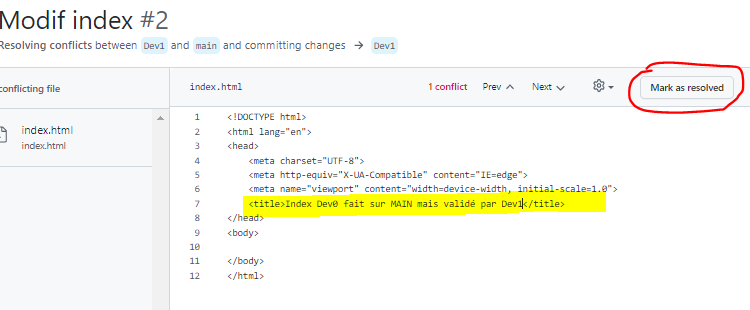
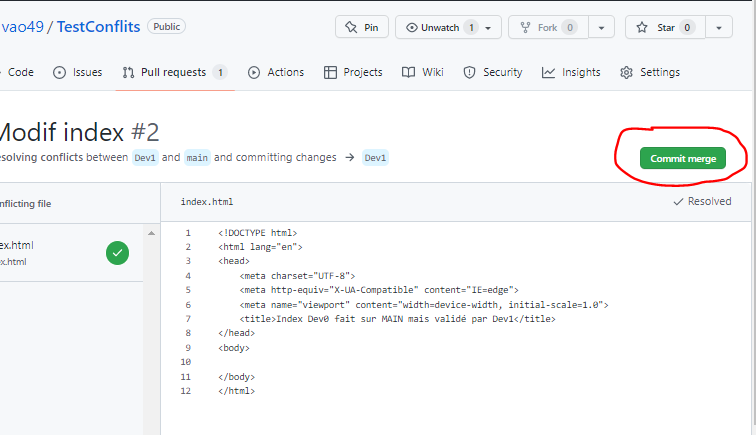
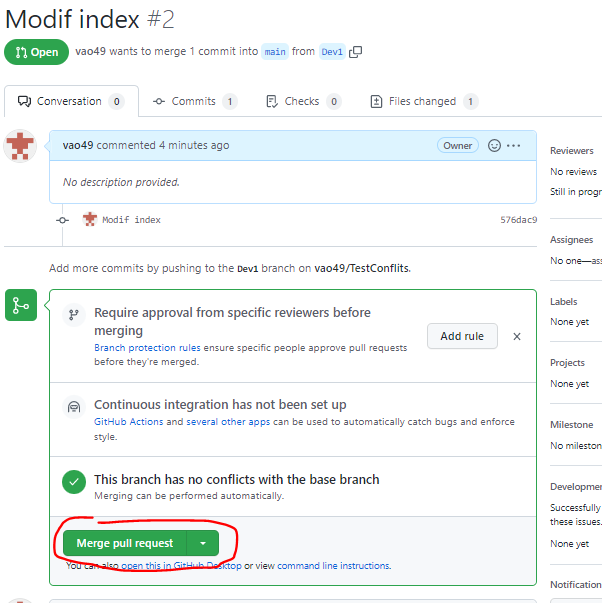
# Etat initial :

* Dev0 a créé un projet sur GITHUB avec un fichier index.html
* Dev1 va cloner ce projet et faire une branche Dev1
* Dev1 travaille sur cette branche en modifiant le contenu du fichier index.html comme ceci



* Parallèlement Dev0 fait aussi une modification sur index.html dans la branche main et PUSH le tout sur GITHUB
* Ensuite Dev1 fait un PUSH vers GITHUB
* Dev0 va générer la PULL REQUEST
* Puis
* GitHub va indiquer qu’il y a des conflits 🡺 Il faut les résoudre en cliquant sur le bouton
* Sur chacun des fichiers, nous allons pouvoir identifier les conflits et les résoudre
* Une fois la modification dans le code effectuée, il faut « marquer » ce conflit comme résolu
* Une fois que tous les conflits sont résolus il faut COMMIT/MERGE
* GITHUB va réanalyser tout le code pour vérifier qu’il n’y a plus de conflit et vous proposer de MERGER

Et voilà les conflits sont résolus.

Il ne vous reste plus qu’a faire le ménage (suppression de la branche Dev1) et faire un PULL sur MAIN en local

# Autre possibilité (résolution des conflits en LOCAL par Dev1)

* Une fois la PULL REQUEST créée et indiquant un conflit
* Dev1 est informé via la messagerie d’un conflit
* Dev1 fait un PULL sur la branche de base (main) de sa branche de développement (Dev1)
* Dev1 fait un MARGE de main sur la branche Dev1 🡺 Dev1 verra l’ensemble des conflits et pourra les corriger
* Dev1 fait un COMMIT et PUSH sur Dev1
* Normalement (si tous les conflits on été résolus par Dev1), la PULL REQUEST devrait passer en VERT et Dev0 pourra MERGER