TP4

Corrigez vos erreurs sur un dépôt local

1) Créer un dossier "Test" avec la commande ***mkdir***, qui sera notre dépôt Git, et dirigez-vous dans ce dossier.
2) Initialisez le dépôt.

1ère erreur effectuée à corriger : Créer une branche par erreur

1) Avant de créer une branche, vous devez créer votre branche principale.
   Pour cela, il vous suffit de créer un fichier "PremierFichier.txt" dans votre répertoire Test, l'ajouter à l'index et le commiter.
2) Si votre branche principale se nomme master, nommer la « main ».
3) Créer une nouvelle branche autre que la branche principale et nommer la « brancheTest »
4) Vérifier que la branche a bien été créée en affichant les branches existantes.
5) Nous voulions ajouter nos fichiers avant de la créer et nous sommes maintenant bloqués avec cette branche que nous ne voulions pas tout de suite. Supprimez alors cette branche « brancheTest ».
6) Vérifier que la branche a bien été supprimée en affichant les branches existantes.

2ème erreur effectuée à corriger : Modifier la branche principale (sans commiter)

Si vous avez modifié votre branche principale (main ou master) avant de créer votre branche et que vous n'avez pas fait le commit, ce n'est pas bien grave. Il suffit de faire **une remise** - ou un stash en anglais.

La remise, ou stash, permet de mettre les modifications de côté, les ranger, le temps de créer la nouvelle branche et d'appliquer cette remise sur la nouvelle branche.

1) Aller sur la branche principale pour modifier des fichiers. Par exemple, créer un fichier « fichier.txt » dans le dossier que vous avez déjà créé « Test », puis ajouter le à l'index.
2) Vous pouvez à tout moment voir l'état dans lequel sont vos fichiers, c'est-à-dire voir les changements qui ont été indexés ou ceux qui ne l'ont pas été, avec la commande ***git status***. Afficher l'état d vos fichiers.
3) Créer un stash.
4) Assurez-vous que la branche principale soit de nouveau propre, en faisant un nouveau ***git status.***
5) Créer une nouvelle branche « brancheCommit ».
6) Afficher les branches existantes.
7) Basculer sur cette branche.
8) Appliquer le stash afin de récupérer les modifications que vous avez rangées dans le stash et Appliquer ces modifications sur votre nouvelle branche.
9) Supposons que vous avez créé plusieurs stash, et que le dernier n'est pas celui que vous souhaitez appliquer. Utiliser la commande permettant d'afficher la liste de vos stash.
10) Appliquer cette fois, le premier stash.

3ème erreur effectuée à corriger : Modifier par erreur une branche et réaliser un commit

Admettons que vous ayez réalisé vos modifications et qu'en plus vous ayez fait le commit. Le cas est plus complexe, puisque vous avez enregistré vos modifications sur la branche principale, alors que vous ne deviez pas.

1) Ajouter un nouveau fichier « fichier 2 » dans la branche « brancheCommit », ajouter à l'index et réaliser le commit.
2) Pour réparer cette erreur, vous devez analyser vos derniers commits avec la fonction ***git log*** . Vous allez alors récupérer l'identifiant du commit que l'on appelle couramment le *hash*. Par défaut, ***git log*** va vous lister par ordre chronologique inversé tous vos commits réalisés.

3) Maintenant que vous disposez de votre identifiant, gardez-le bien de côté. Vérifiez que vous êtes sur votre branche principale et réalisez la commande suivante :

```
git reset --hard HEAD^
```

Cette ligne de commande va supprimer de la branche principale votre dernier commit. Le HEAD^ indique que c'est bien le dernier commit que nous voulons supprimer. L'historique sera changé, les fichiers seront supprimés.

4) Créez une nouvelle branche « nouvelleBranche ».
5) Basculer sur cette branche.
6) Maintenant, vous êtes sur la bonne branche. Renouvelez la commande ***git reset***, qui va appliquer ce commit sur votre nouvelle branche.

4ème erreur effectuée à corriger : Message du commit erroné

Lorsque l'on travaille sur un projet avec Git, il est très important de marquer correctement les modifications effectuées dans le message descriptif.

Cependant, si vous faites une erreur dans l'un de vos messages de commit, il est possible de changer le message.

1) Imaginez que vous veniez de faire un commit et que vous ayez fait une erreur dans votre message. Ecrire la commande vous permettant de modifier le message de votre dernier commit.
2) Vérifier que le message a été ajouteé avec la commande ***git log***.

5ème erreur : Fichier oublié à ajouter dans le dernier commit

1) Créer un fichier « FichierOublie » et ajouter le à l'index.
2) Imaginez que vous ayez fait votre commit mais que vous réalisiez que vous avez oublié ce fichier. Ecrire la commande qui vous permet d'ajouter le fichier au dernier commit.