

UE1 CONS : CONSOLIDATION BASES DE DONNÉES ET PROGRAMMATION OBJET.

VERSIONNAGE ET TESTS.

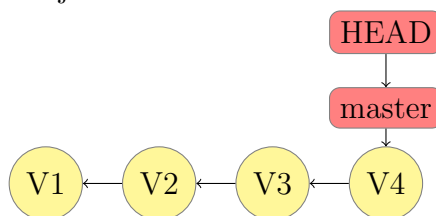
Cours-TD n°3

Introduction à git

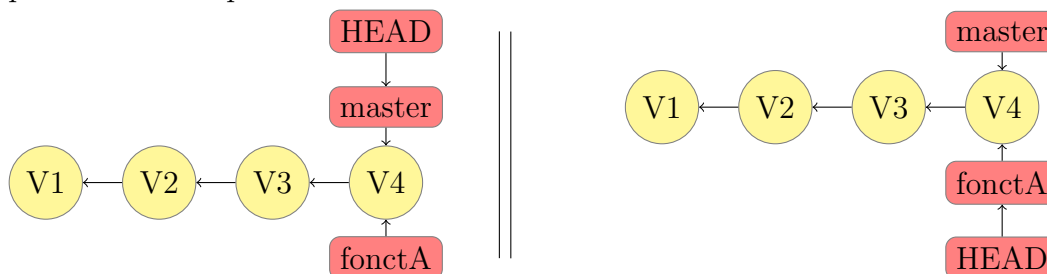
**Cette feuille va aborder l'utilisation des branches dans un dépôt git.**

Supposez que vous ayez un projet opérationnel que vous voulez compléter en ajoutant une fonctionnalité. Pendant que vous développez votre nouvelle fonctionnalité, on vous prévient qu'un bug doit être corrigé dans la partie initiale de votre projet. Vous travaillez sur la correction de ce bug, une fois terminé, vous apportez cette correction sur le projet opérationnel puis vous continuez à mettre au point la nouvelle fonctionnalité voulue. Une fois terminée, vous l'intégrez à votre projet. Les branches git vont vous aider à gérer proprement ce processus de développement.

1. Projet initial :



2. Vous avez une nouvelle fonctionnalité à développer. Pour conserver votre projet opérationnel, vous allez créer une nouvelle branche (comme si vous faisiez une copie de votre projet) avec la commande `git branch fonctA` puis vous allez placer vous répertoire de travail et votre index sur cette branche : `git checkout fonctA`. (ou en une seule commande `git checkout -b fonctA` qui crée la branche si elle n'existe pas avant de se positionner dessus.



3. Travaillez sur votre nouvelle fonctionnalité. Ajoutez un fichier fonctionA.txt. Versionnez.

```
1 echo "Début du code de la fonctionnalité A" >> fonctionA.txt
2 git add fonctionA.txt
3 git commit -m"Debut de fct A"
4 git push
```

Le `git push` ne va pas fonctionner car le `push` est programmé pour être fait sur la branche `master`. Pour pousser votre branche sur le dépôt distant vous devez vous mettre sur la branche `master` puis pousser votre nouvelle branche.

```
1 git checkout master
2 git push origin fonctA
```

4. Continuez de développer votre fonctionnalité. Versionnez.

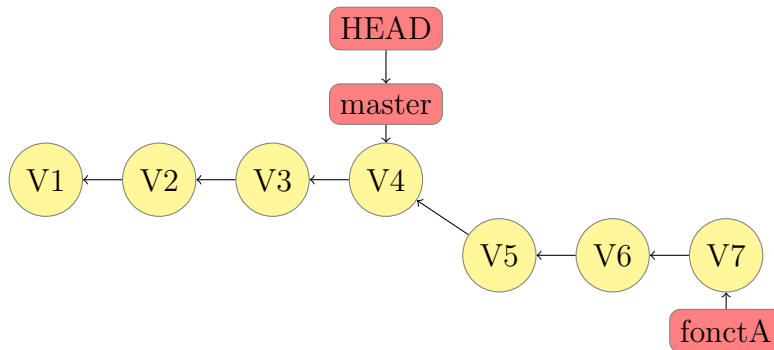
```
1 git checkout fonctA
2 echo "Code complété de la fonctionnalité A" >> fonctionA.txt
3 git add fonctionA.txt
4 git commit -m"Fonct A complétée"
5 git checkout master
6 git push origin fonctA
```

Vous pouvez vérifier la présence de vos fichiers et de votre branche sur [gitlab.com](https://gitlab.com).

5. Terminez votre fonctionnalité. Versionnez.

```
1 git checkout fonctA
2 echo "fonctionnalité A terminée" >> fonctionA.txt
3 git add fonctionA.txt
4 git commit -m"Fonct A terminée"
5 git checkout master
6 git push origin fonctA
```

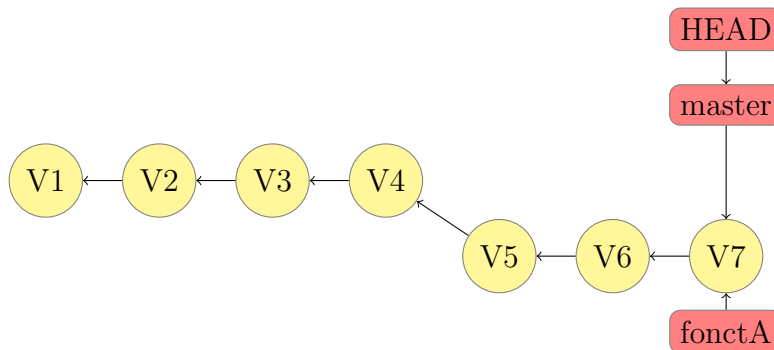
Vous êtes dans cette situation :



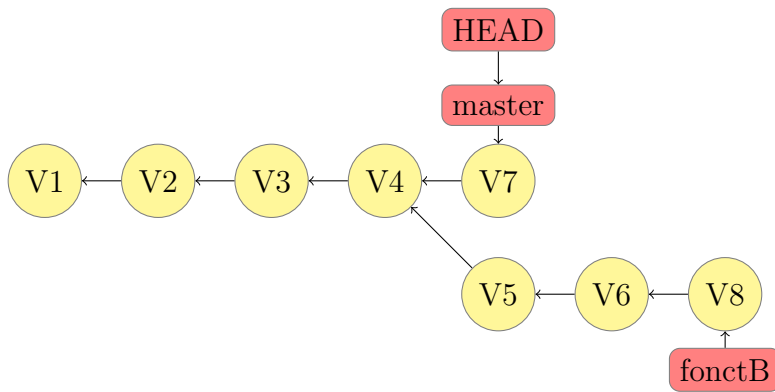
6. Intégrez votre travail dans votre projet principal. Pour cela vous allez fusionner votre branche **fonctA** à votre branche principale puis pousser votre nouvelle version sur le dépôt distant.

```
1 git checkout master
2 git merge fonctA
3 git push
4 git branch -d fonctA
```

La dernière commande permet de supprimer la branche **fonctA**. Vous avez cette situation :



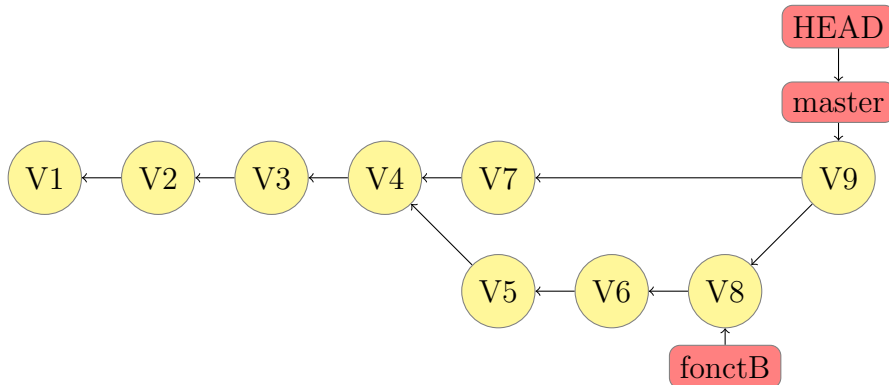
7. Créez une nouvelle fonctionnalité dans une nouvelle branche mais avant de l'avoir terminée, créez une nouvelle version de votre projet principal (qui correspondrait à la correction d'un bug par exemple). Terminez votre nouvelle fonctionnalité. Vous devez avoir cette situation :



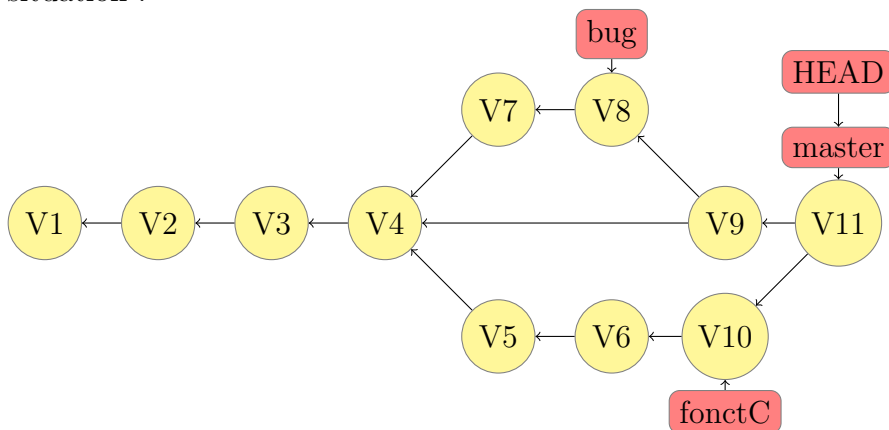
8. Intégrez votre nouvelle fonctionnalité à votre branche principale.

```
1 git checkout master
2 git merge fonctB
3 git push
```

Cela ne va pas poser de problème car il n'y a pas de conflit, il va y avoir création d'une nouvelle version avec la confirmation du message associé (Merge branch 'fonctB'). Nouvelle situation :



9. Pour corriger le bug, vous auriez pu créer une autre branche et obtenir ce genre de situation :



Pour cela vous devez créer une branche **fonctC**, travailler sur cette branche avec deux commits, puis créer une branche **bug** pour corriger l'anomalie rencontrée avec deux commits aussi pour avoir la solution. Vous fusionnez **bug** dans **master** `git merge bug` sur la branche **master**, vous pouvez supprimer la branche **bug**. Vous continuez à développer la fonctionnalité C sur la branche **fonctC**, une fois terminé, vous incorporez cette nouvelle fonctionnalité dans votre branche **master** avec `git merge fonctC`. (Pensez à toujours pousser sur le dépôt après merge ou commit).