

**RAPPORT DE SCENARIO #3**

**Réalisée par : Nouhaila iounousse, MASSIF Abdessamed, Nada ElFadili, Hassan Hmiyed ,Zakia Antary et abdelhaq elfakir**

**Encadré par : Mr. Youssef chouni**

**Année universitaire : 2019/2020**



*Rapport*

# Scénario #3

*Product BACKLOG :*

**Step 1: Changes on GitHub**

* **Sur votre compte Github : faites les modifications suivantes :**
* **Sur le dossier CSS ajoutez un autre fichier nommée style.css**



* **Sur index.html ajoutez juste après la balise <body> une balise <H1> : <h1> modification récente </h1>**

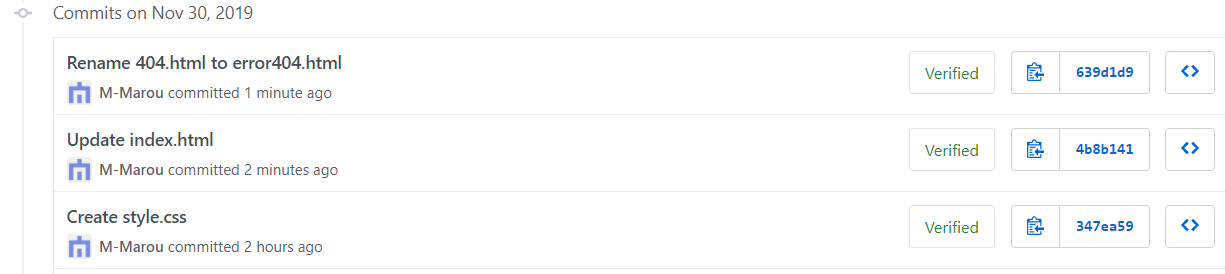


* **Renommer le fichier 404.html en error404.html.**

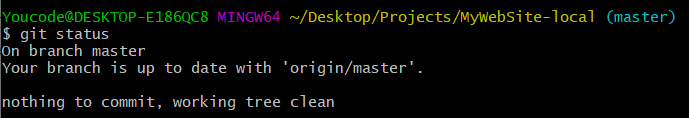


* **Maintenant la synchronisation avec votre repository local, les changements effectués doivent impérativement apparaitre sur votre repo LOCAL :**

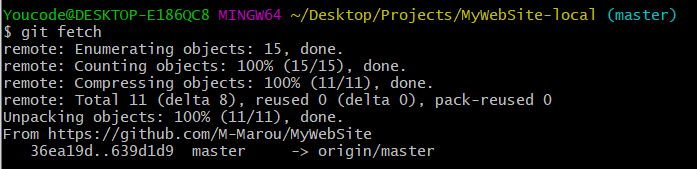
1. **Sur gitHub vérifiez la liste des commîts.**

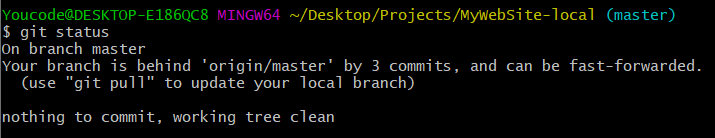


1. **Sur GITBASH, vérifiez avec « git status » .**

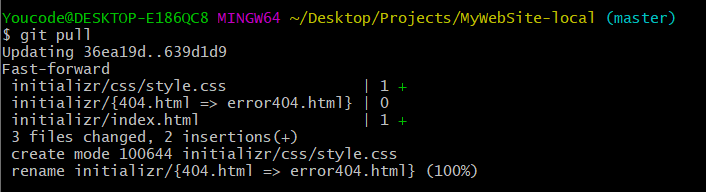


1. **Visualisez et télécharger les fichiers distants sur GITBASH.**

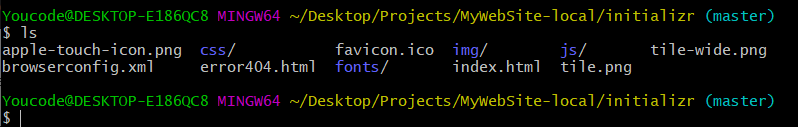




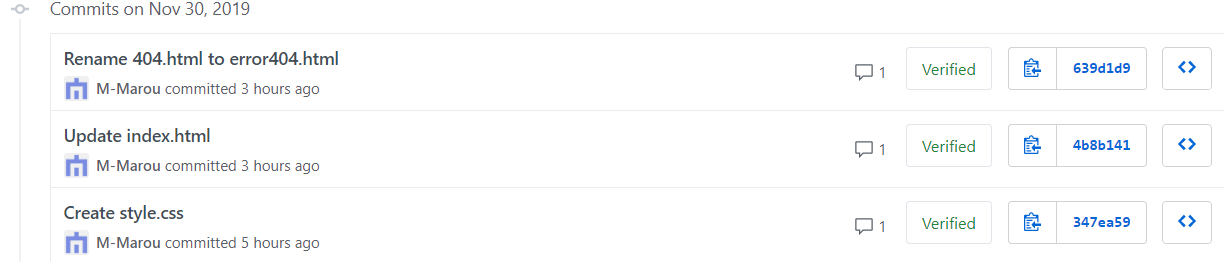
1. **Maintenant faites le pull et fusionnez les changements distants avec le repo local.**



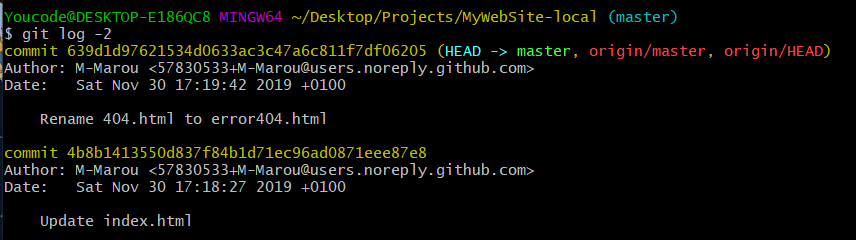
1. **Vérifiez votre repo Local.**



1. **Sur votre repo Distant GITHUB, laissez un commentaire sur les fichiers que vous avez modifiez en regardant la liste des commits.**



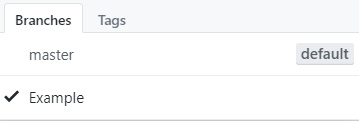
1. **Sur GITBASH affichez les informations du commit de l’ajout de la balise <h1></h1>.**



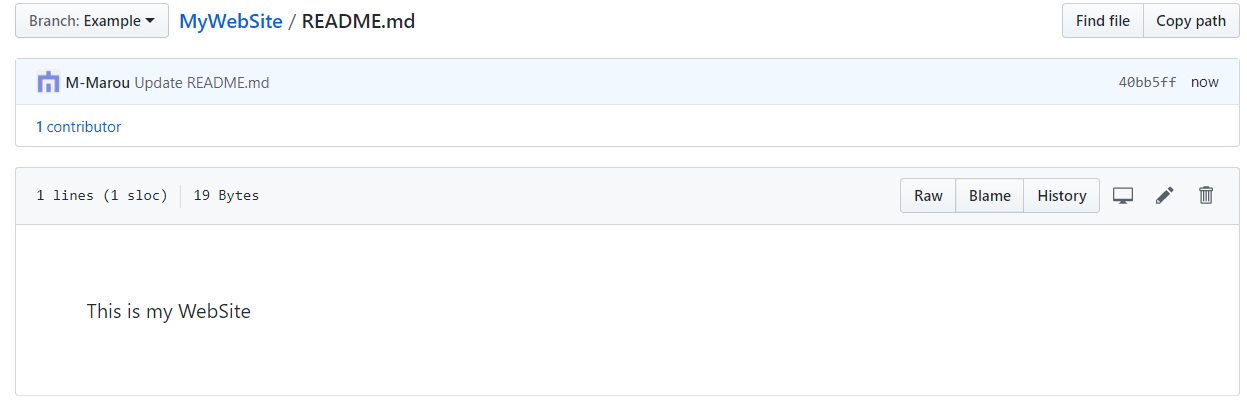
**Step 2: Branching and merging sur GITHUB**

* **1. Repo distant :**

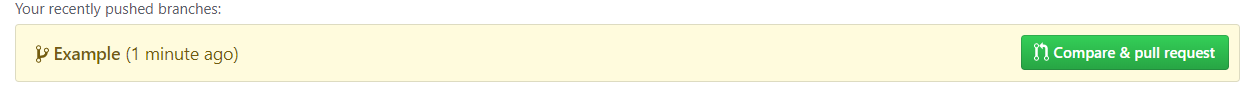
1. **Sur GitHUB créez une branche du nom ‘Example’**



1. **Sur la branche ‘Example’, modifiez le fichier READM.md et faites-le commit (toujours sur GITHUB)**

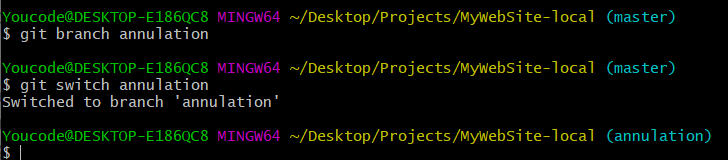


1. **Sur github et sur le menu ‘branch’ ; Expliquez ce que vous voyez comme message ?**

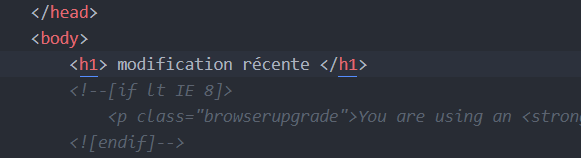


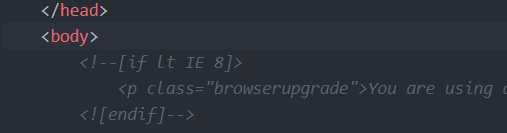
* **2. Repo local :**

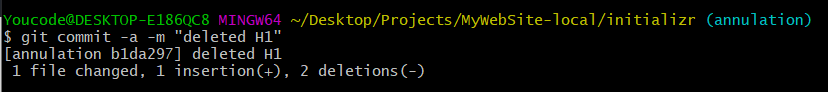
1. **Sur votre repo local, créez une branche nommé ‘annulation’.**



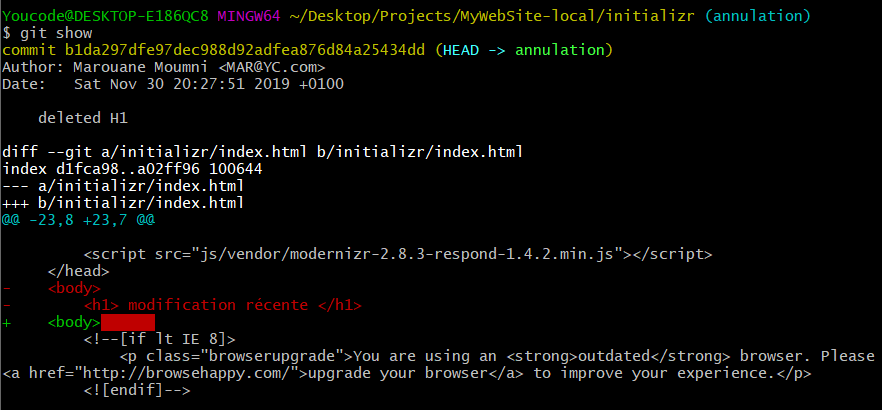
1. **Sur votre repo local éditez le fichier « index.html » et supprimez la balise H1.  Faites-le commit et le stagging en une seule ligne.**





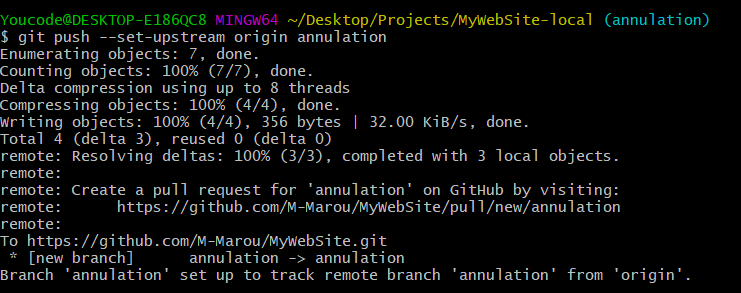


1. **« Git show » pour vérifier vos changements.**

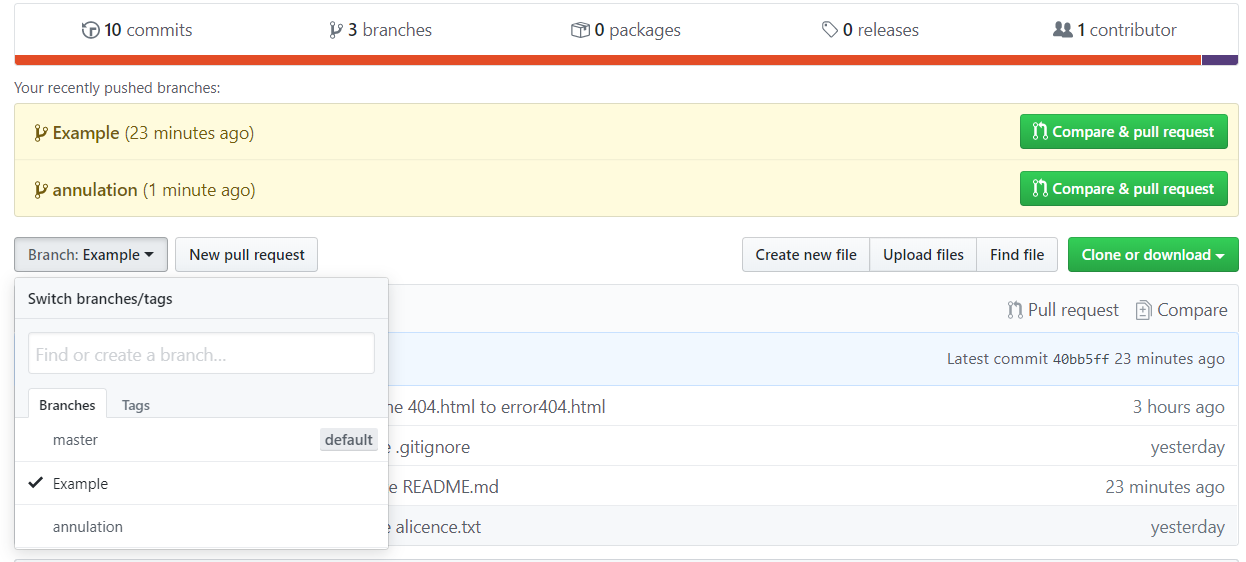


* **Vérification :**

1. **Sur GITBASH Pushez les changements que vous avez sur la branche ‘annulation’ à votre repo distant**



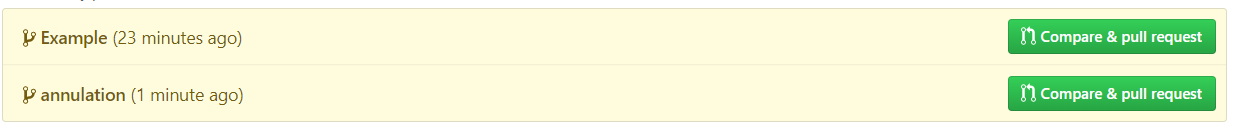
1. **Maintenant vérifiez votre repo distant.**



1. **Qu’est-ce que vous constatez ?**

**Step 3: Compare pull Requests**

* **Sur votre repo Distant, un message indiquant qu’une branche a été ajoutez avec un bouton « compare and pull request »**

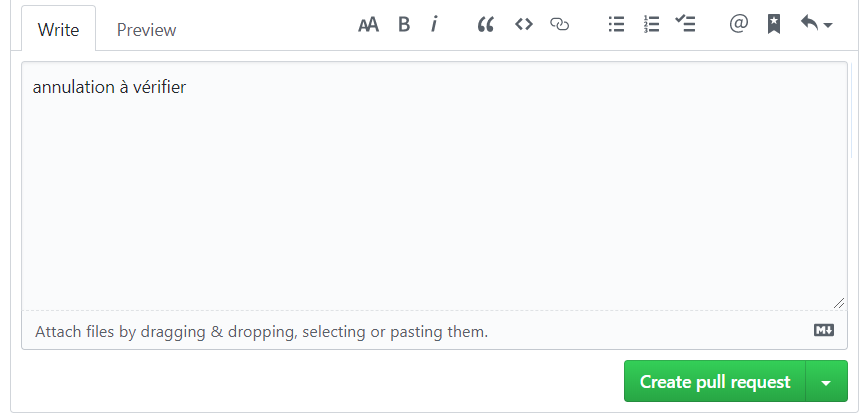


* **Cliquez sur le bouton ? analysez le rendu de la page ? qu’est-ce que vous constatez ?**

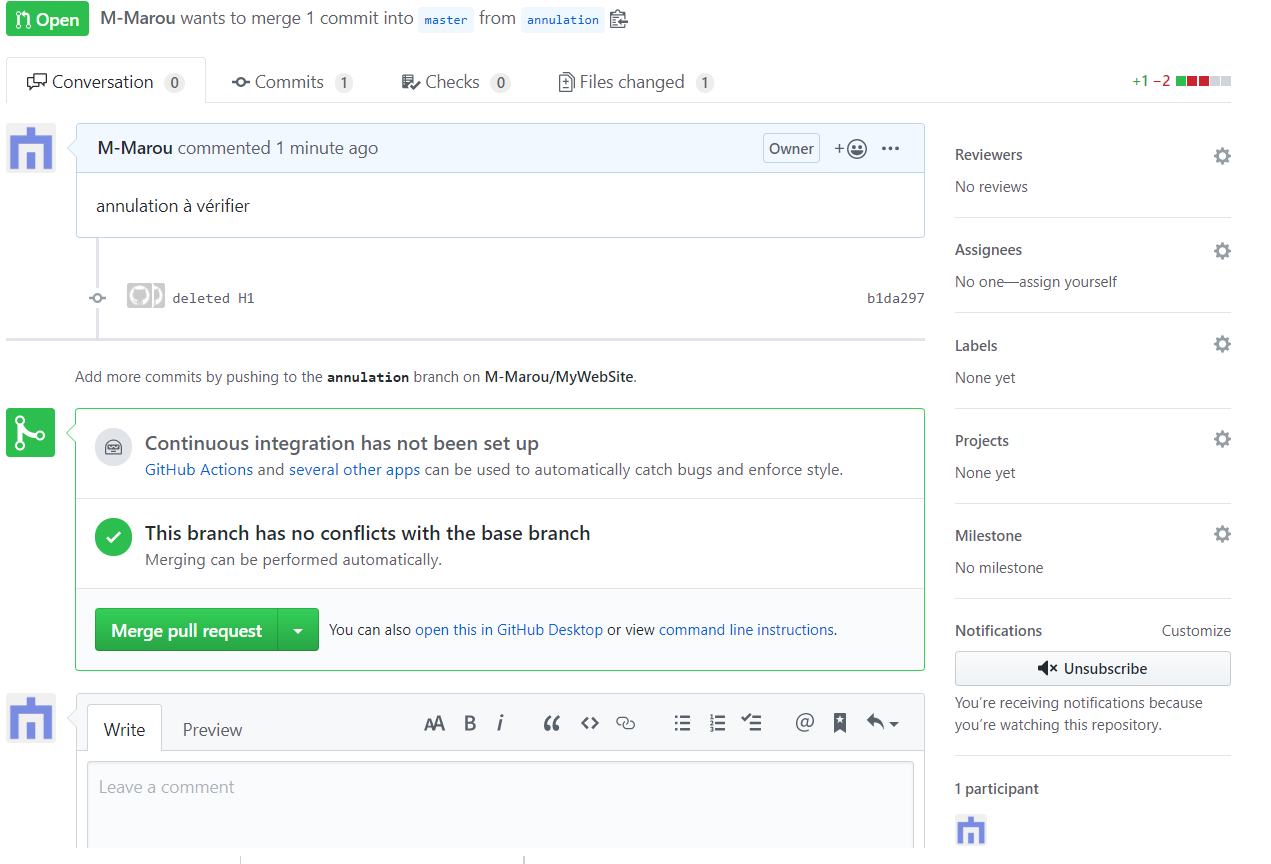


**-il montre ce qui a été changé**

* **Créez un pull request avec le commentaire ‘annulation à vérifier’**



* **Une page s’ouvre, explorez la page ? qu’est-ce que vous constatez ?**



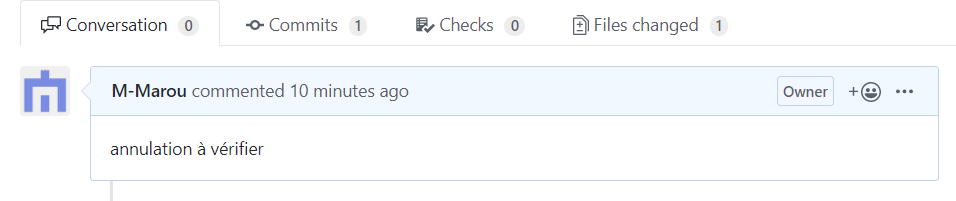
**Après avoir cliqué sur Pull Request et laissé un commentaire, cette page sera chouette**

**et vérifiera si nous avons des conflits avec la branche à laquelle nous souhaitons fusionner.**

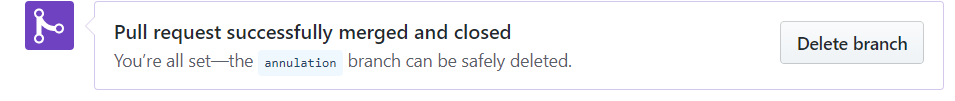
**Généralement, quelqu'un d'autre devrait faire la demande de fusion, mais dans notre cas,**

**si vous travaillez en solo, vous le faites vous-même.**

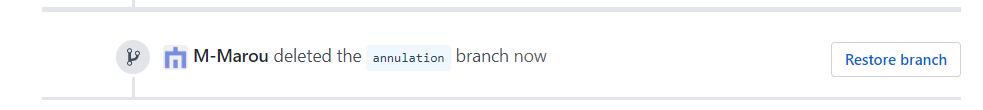
* **Sur l’onglet ‘conversation’ vous pouvez aussi laisser des commentaires à la personne qui demande le pull request ; laissez un commentaire ?**



* Maintenant si tout va bien ; aucun message d’erreur effectuez le merging ?

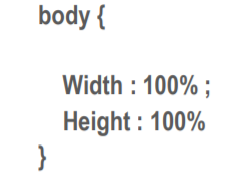


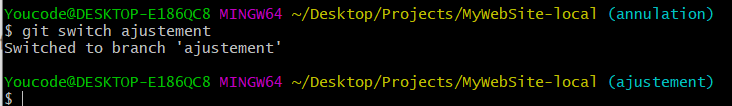
* **Supprimez la branche ‘annulation’ une fois tout est OK.**

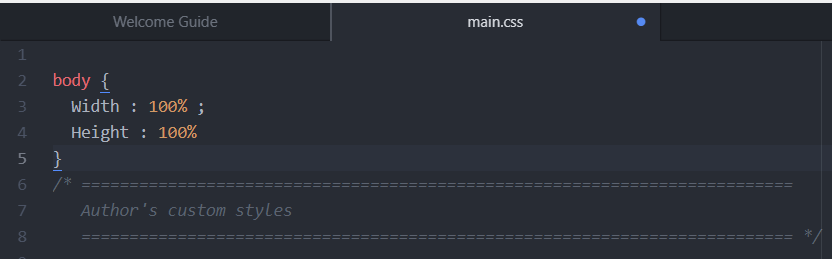


**Step 4 : Merging en local**

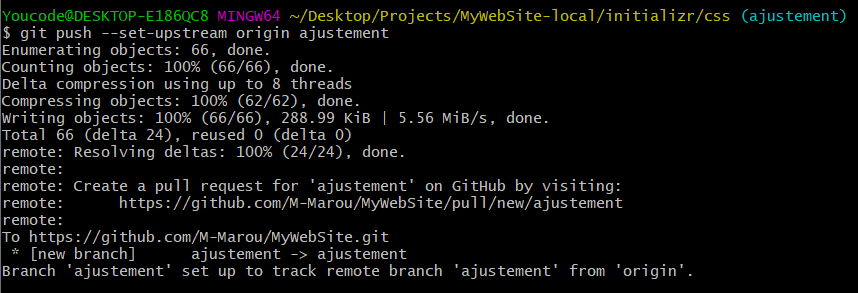
* **Pour finaliser l’objectif de cette étape, on va créer une branche nommé ‘ajustement’, ou on va modifier le fichier ‘main.css’ ( css/main.css) en ajoutant la ligne suivante :**





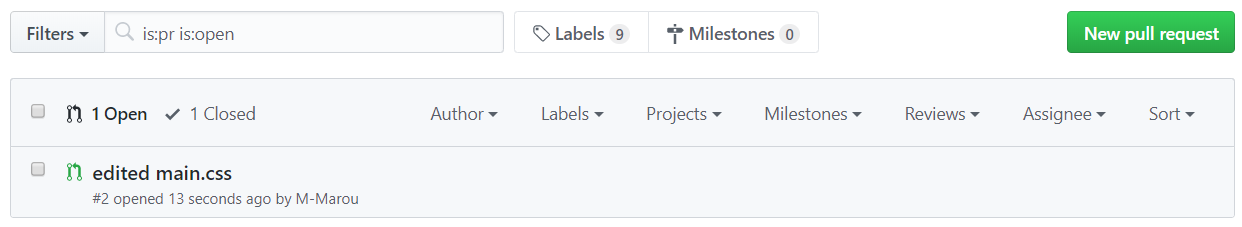


* **Faites maintenant le nécessaire pour ‘pushez’ les modifications sur GITHUB.**



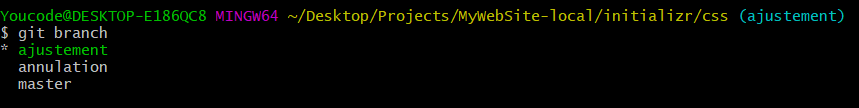
* **Sur votre GITHUB, un pull request est demandé !**



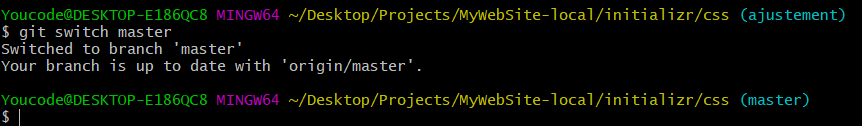


* **Maintenant, essayez de faire le merge en local ! @ vos mains !**

1. **Sur GITBASH, Vérifiez dans quelle branche vous êtes maintenant ! surement sur la branche ‘ajustement’**

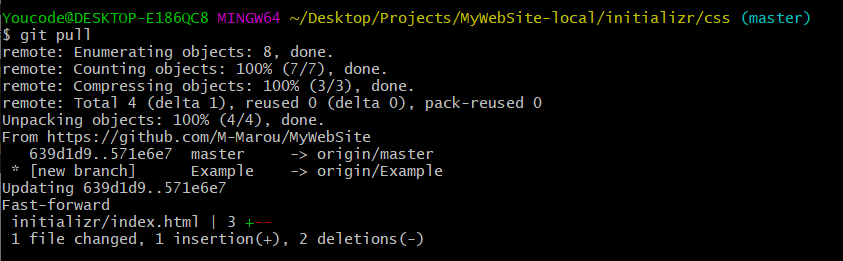


1. **Un pull Request est demandé sur GITHUB ! N’oubliez pas !**
2. **Sur GITBASH, revenez vers la branche master ! Expliquez le rendu de la commande ?**

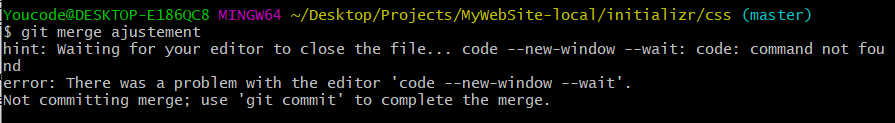


**-Retour à la branche principale pour faire une demande de tirage**

**Maintenant demandez le pull ?**



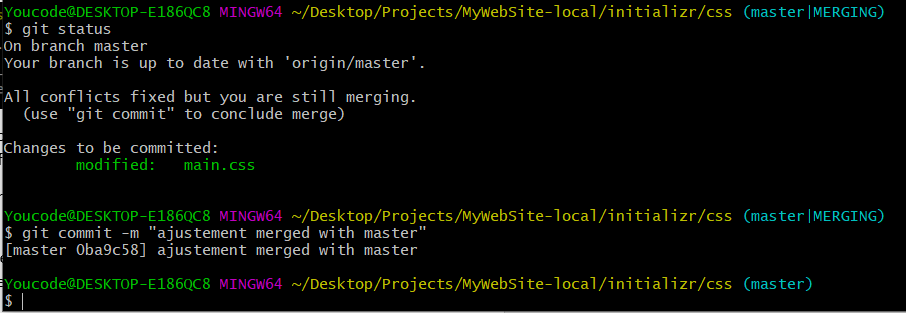
1. **Une fois la commande du pull est exécutée, à votre avis quel type de merge on aura besoin pour faire un merge sans conflits**
2. **Faites le merge à travers GITBASH**



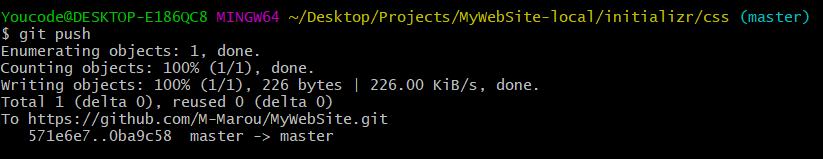
1. **Une fenêtre s’ouvre après l’exécution de la commande du merge ; ce qui signifie le message délivré ? à quoi sert cette fenêtre ?**

L'éditeur de texte s'ouvre pour nous permettre de vérifier les modifications et de corriger les conflits.

1. **Git status**



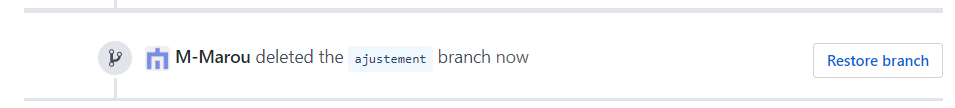
1. **Si tout est OK ! Pushez les changements ?**



1. **Si tout est OK ! sur GITHUB le pull sera vérifier ? explorez le pull pour visualiser les changements ?**

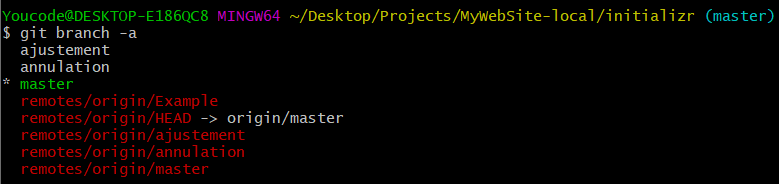
ok

* Sur GITHUB supprimez les branches que vous avez créé.



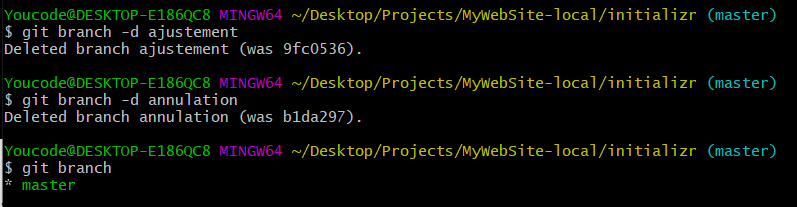
* **Sur le local on doit apporter les changements qu’on a fait sur GITHUB :**

1. **Tapez la commande git branch –a ? expliquez le message ?**

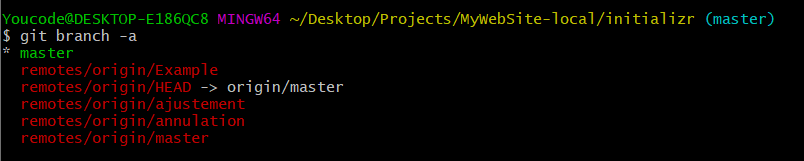


**-Cette commande affiche la liste de toutes les branches**

1. **Sur GITBASH supprimez les branches que vous avez créé.**

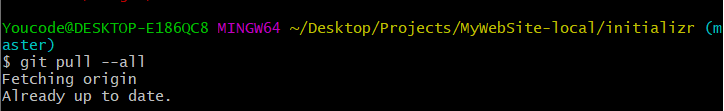


1. **Maintenant ! Tapez la commande git branch –a ? qu’est-ce que vous constatez ?**



**-Cette commande affiche la branche dans laquelle je me trouve et la liste de toutes les branches supprimées.**

1. **Ce que nous devons faire maintenant, est de faire appelez la suppression faites sur GITHUB et l’appliquez sur le local ! Trouvez la commande et exécutez ?**

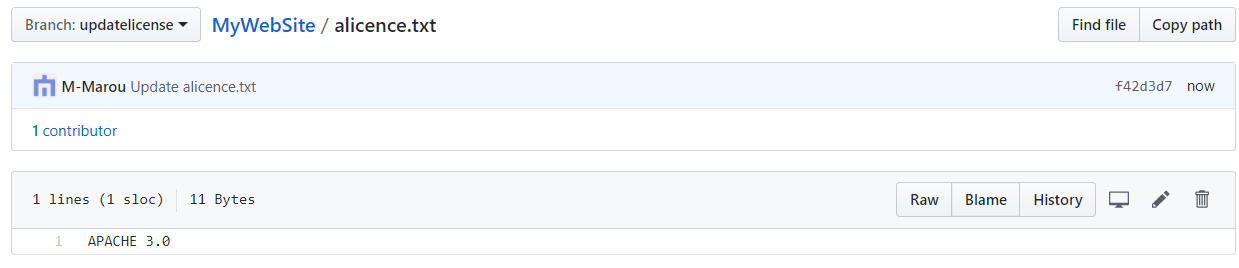


1. **Tapez la commande git branch –a ? qu’est-ce que vous constatez ?**

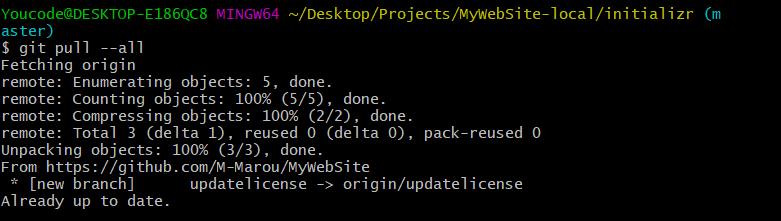
**Step 5 : The Cleaning up**

* **Dans cette phase je vous laisse découvrir l’objectif ?**
* **Sur GITHUB Créez une branche nommée ‘updatelicence, éditez le fichier Licence et changer Apache 2.0 par Apache 3.0**

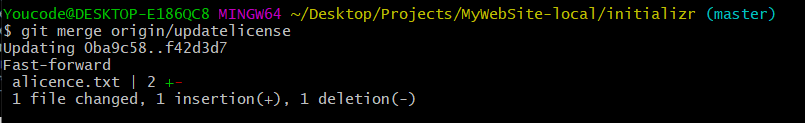




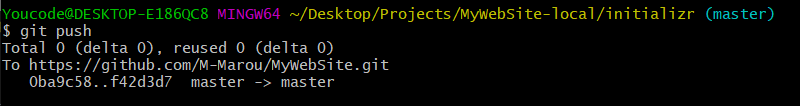
* Sur GITBASH faites un pull global : git pull –-all ;



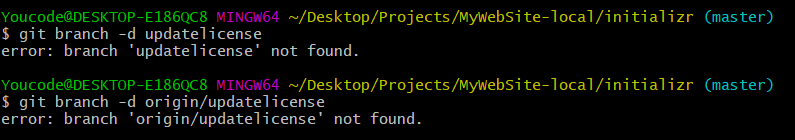
* **Maintenant faites le merge de la branche ‘updatelicence’.**



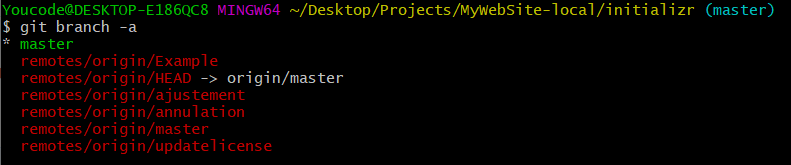
* **Exécutez Git push !**



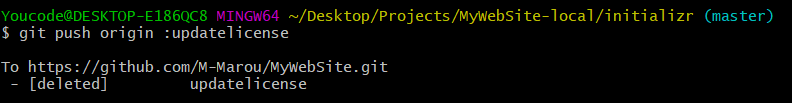
* **Git Branch –d ‘updatelicence’**



* **Exécutez la commande git bash –a ? qu’est-ce que vous constatez sur le remote et aussi sur GTHUB?**



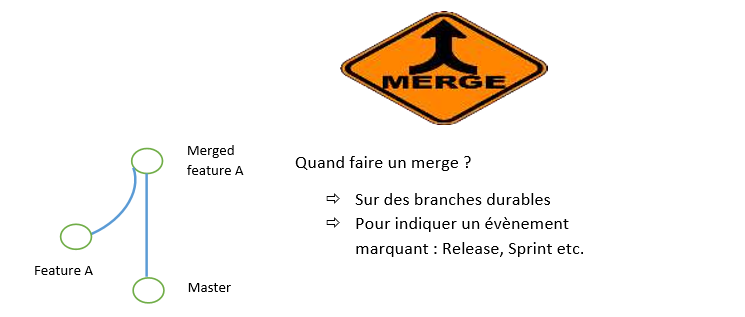


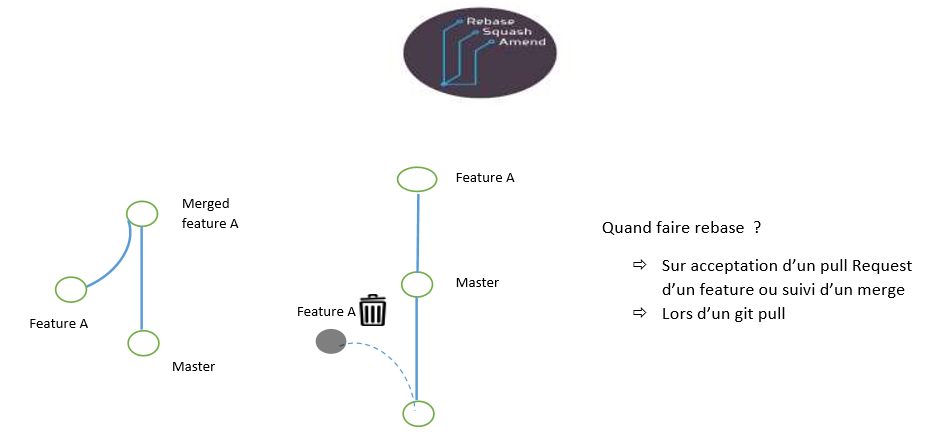


After we delete a branch localy it still exist on the remote, so the objective was to find an command capable of deleting the branch on the local and the remote also.

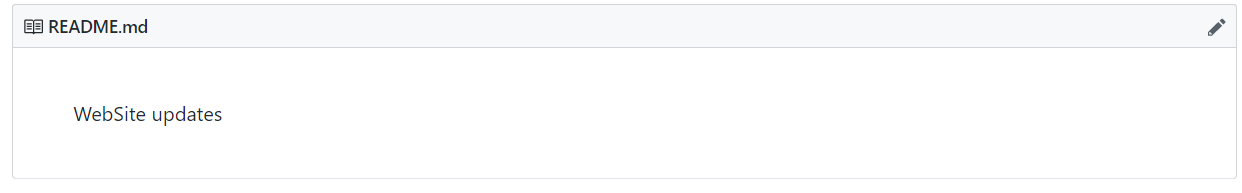
**STEP 6 : Rebasing**

* Introduction générale : Rebase vs merge

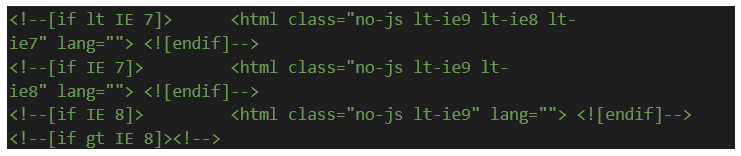




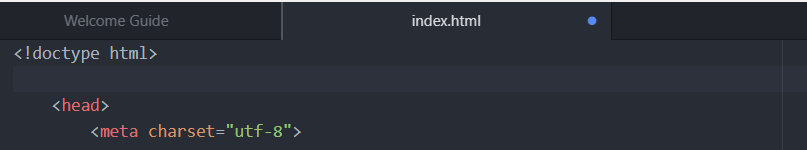
* **Sur GITHUB, Editez le fichier README.md, ajoutez la ligne ‘updates’ ! n’oubliez pas le commit**



* **Sur GITBASH Editez le fichier index.html supprimez les lignes suivantes :**



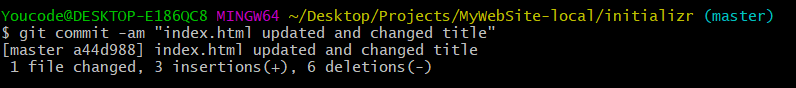




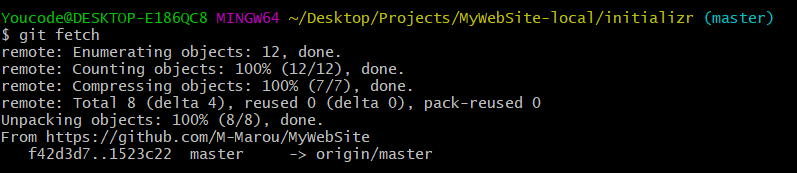
**Et modifiez le titre du site : choisissez le titre que vous vouliez**



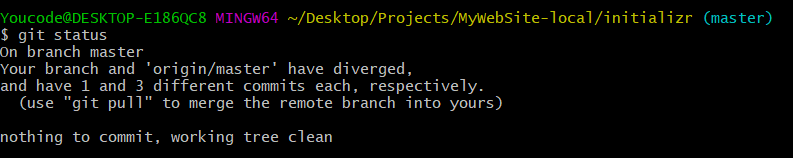
* **Faites le stagging et le commit en une seule ligne**



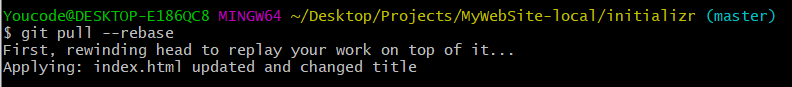
* **Maintenant apportez les changements faites sur le Repo distant : fetch**



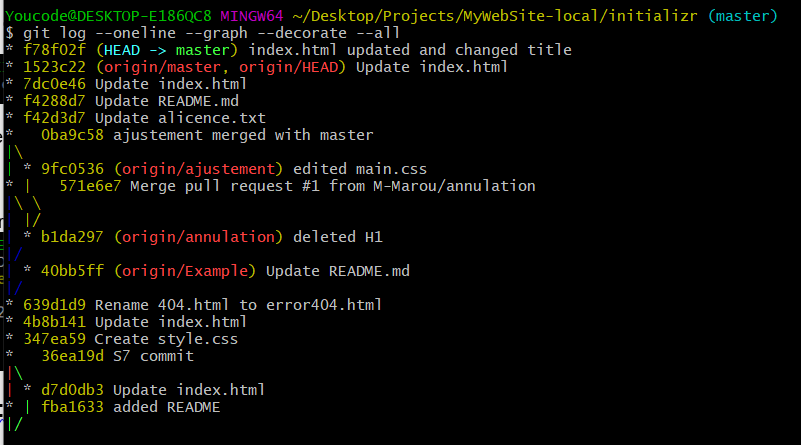
* **Git status ? lisez les commentaires**



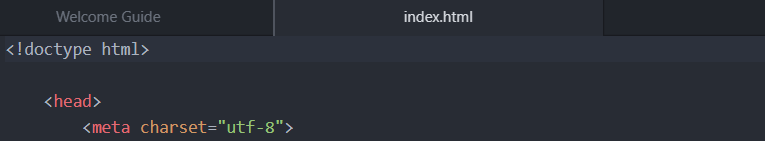
* **Maintenant si vous compreniez le commentaire, on a besoin de faire un rebase : git pull - - rebase**

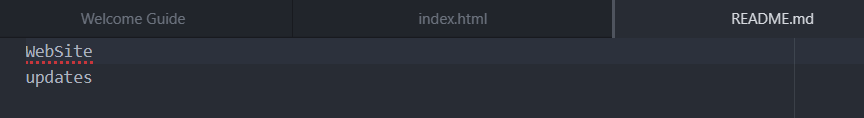


* **Exécutez l’alias que vous avez créé dans le scénario #1 (historique) ? qu’est-ce que vous constatez sur l’arborescence des branches ?**



* **Vérifiez que tous les changements que vous avez faits sont OK ?**





* **Expliquez l’utilité du REBASE ?**

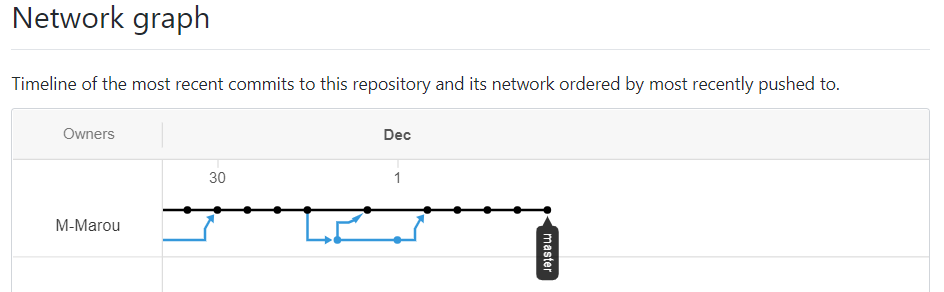
Git rebase does merge and creat a new commit infront of the last commit of the master branch

**Step 7 : GitHub Insights:**

* **Sur Github vous avez un menu INSIGHTS ; Explorez votre activité à travers insights :**

**It has all the statistics of the github repo**

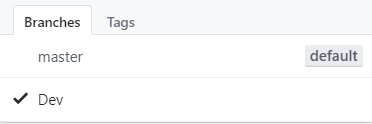
* **Analysez le Network ou le timeline de votre workflow ?**



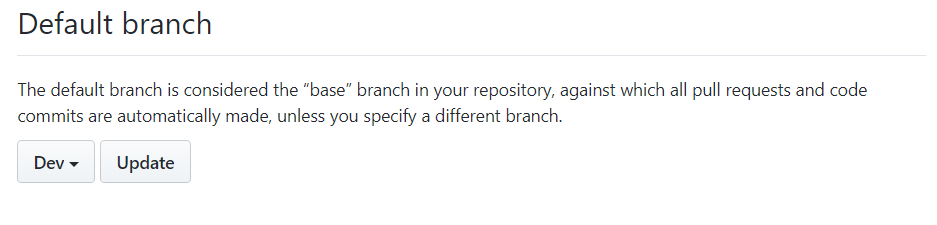
* **Discutez entre vous équipe et formalisez ce que vous voyez sur le Network en une production Timeline sur Tableau à marqueur ou trello : imaginez aussi que chaque membre du groupe à contribuer dans une tache du Timeline Network ( Like a Sprint planning) .**

**Step 7: Default branch and conflicts**

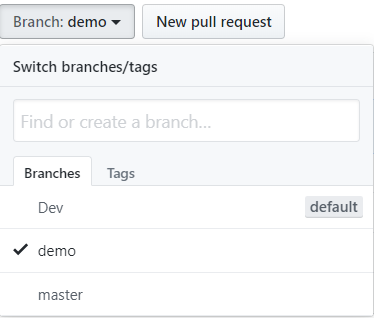
* **Sur github créez une branche du nom “Dev”**



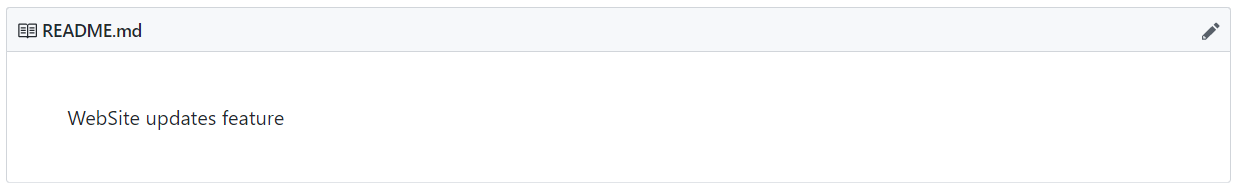
* **La Branche Dev deviendra la branche par défaut ; sur le menu « settings » mettez la branche Dev en mode par défaut**



* **Créez une autre branche « demo », on va la considérer comme « feature ».**



* **Avant de faire un pull Request sur le branche demo , Editez le fichier « README.md » , Ajoutez une ligne**



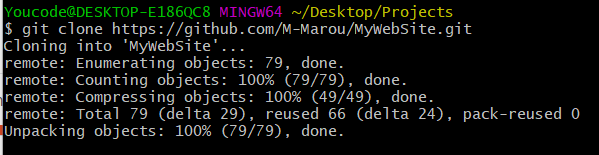
* **Revenez à la page d’accueil de votre repo ? à votre avis on a besoin dans ce cas de faire un merge ou bien un rebase ? pourquoi ?**

**In our case we just changed README file so merge should be enough, but if we working on a big modification on a branch as a feature rebase should be the better choice.**

* **Ne faites aucun merge ni Rebase mais plutôt supprimez la branche feature depuis GITHUB**



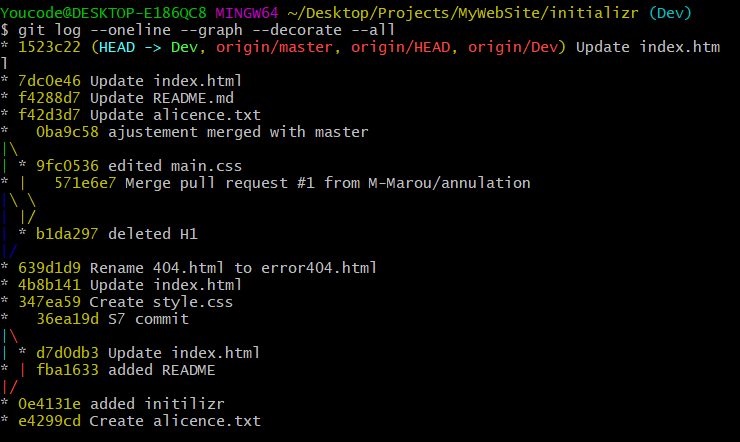
* **Sur GIT, supprimez récursivement et localement votre repo website**
* **Maintenant Clonez via https ou SSH si vous l’avez réussi**



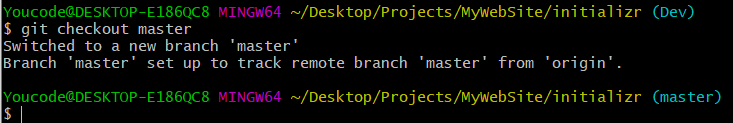
* **Vérifiez si le contenu est récupéré**



* **Vérifiez les branches et ou pointe le pointeur : HEAD**



* **Maintenant on a besoin de la branche master comme base : git checkout master**



* **Toutefois la branche par défaut est : Dev.**