**Réalises par :**

* **ILYASS BENHADDA**
* **SOUKAINA EJJERF**
* **AYOUB BENGARA**

**Rapport pour tous les scénarios**

**LES TACHES:**

1. Scénario #1

1-Immertion

2-Découverte

3-Historique

4-excluding files

5-Branching and Merging

6-conflit résolution

7-mergetools

8- challenge

2. Scénario #2

1- Tagging 2- Stashing and Saving work in Progress:

3- Voyage sur Github, Local Repo to github Repo. 4- Mini challenge.

5- Création d’une local copy. 6- Sending the website :

7- Fetch and pull

3. Scénario #

1- Changes on Github.

2- Branching and merging sur GITHUB.

3- compare pull Requests

4- merging en local

5- The Cleaning up. 6- Rebasing:

7- GitHub Insights:

8- Default branch and conflict

***Scénario #1***

***Step1:***

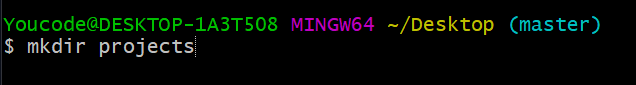
**1 .Déffinition de Git** :

* **Git** est un logiciel de gestion de versions décentralisé.
* C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.
* En 2016, il s’agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes

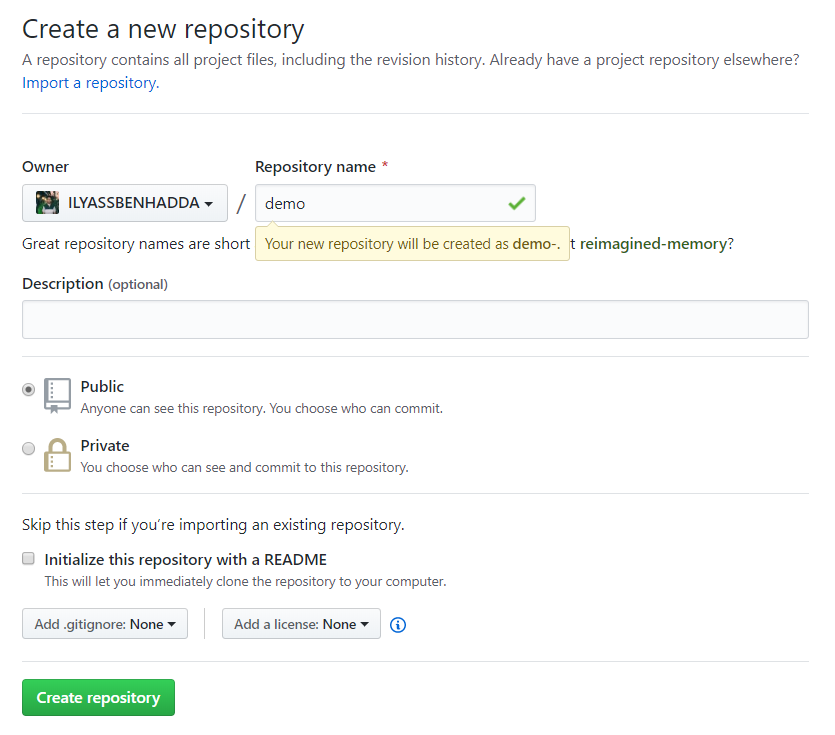
**2. Expliquez le fonctionnement du fichier caché « .git » :**

*.git* créé à l'intérieur du référentiel lors de l'exécution de la commande *git init* . Si vous ne pouvez pas voir le dossier, c'est probablement parce que vos dossiers cachés ne sont pas visibles (utilisez *ls -a* dans un tel cas).. Vous ne l'utiliserez peut-être pas du tout dans votre parcours Git et n'en aurez pas besoin comme tel. Ceci est juste pour vos buts d'apprentissage. Il n'y a pas besoin de l'apprendre ou il n'y a pas de mal à le sauter. En retournant, laissez-le voir à travers le *Git Bash* .

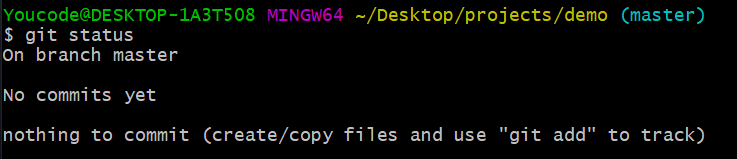
**3. créez un répertoire nommée « /projects ».**

****

**4 .Créez un Repository Local sous le nom « /demo ».**

****

**5. Vérifiez le statut du répertoire « /demo ».**

****

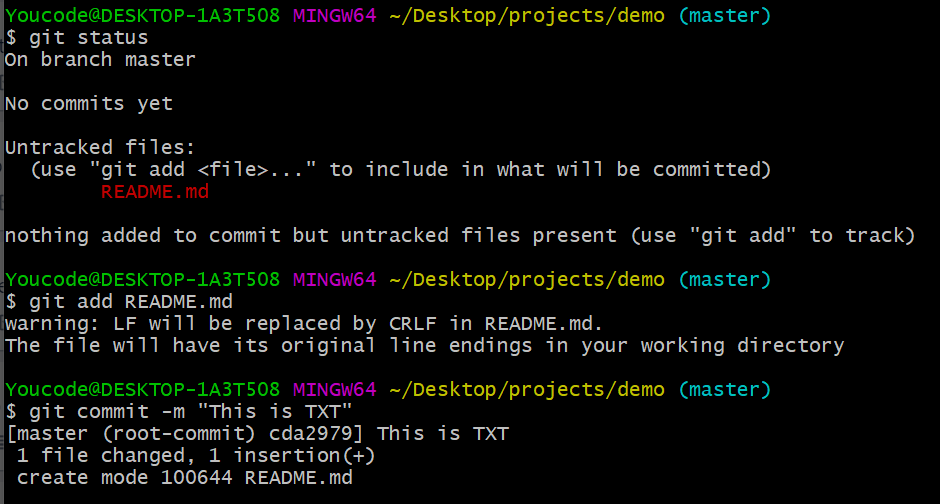
**6. Expliquez le commentaire.**

* Dans bronche master, il a aucun commit yet

**7. Créez le fichier README.md et ajouter la ligne suivante : ‘#Demo project un simple fichier’.**

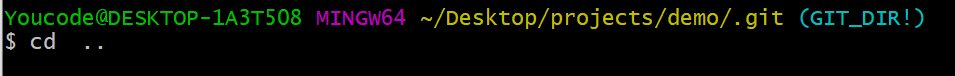
****

**8. Faites le stagging et le commiting avec un commentaire.**

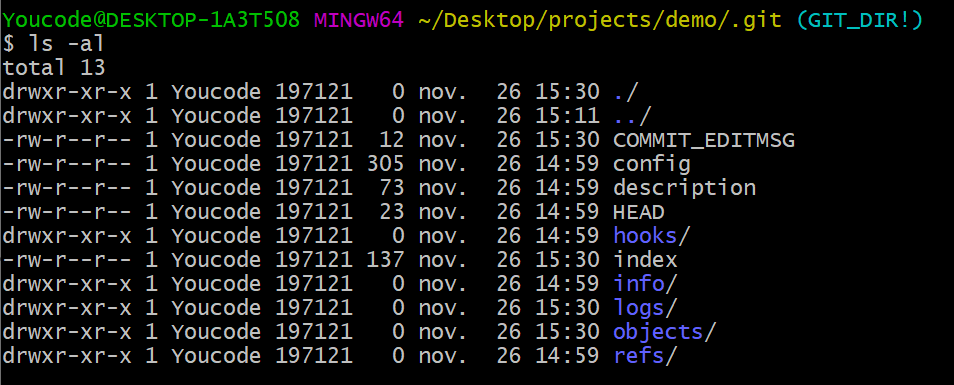
****

***STEP2***

**1 les fichiers de configuration « .git ».**

****

**2 Tapez Ls –al**

****

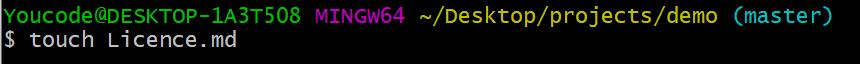
**3 Explication des clauses suivantes : HEAD, LOGS, BRANCHES**

HEAD: Nous sommes sur localisation de branch, dans ce cas en se trouve dans le branche de master (ref: refs/heads/master)

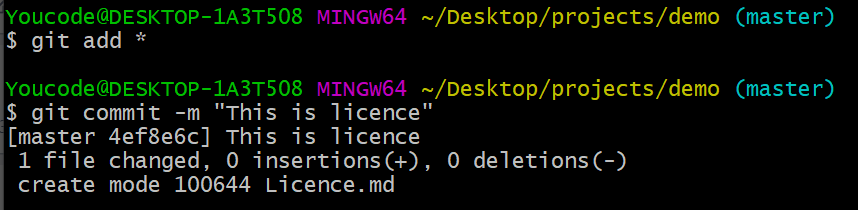
LOGS: entier de git en trouve un fichier qui s’appelle HEAD en trouve les logs des commits .

BRANCHES:( dans ce cas GIT\_DIR!) c’est une localisation d’un fichier “.git “ si ce n’est pas spécifié Git parcourt l’arborescence jusqu’à ou / et cherche un répertoire Git à chaque étape

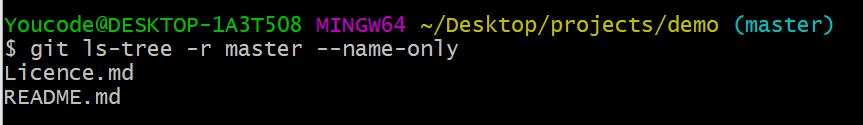
**4 Retournez vers votre Repo et créez le fichier « Licence.md ».**

****

**5 Faites-le Commit**

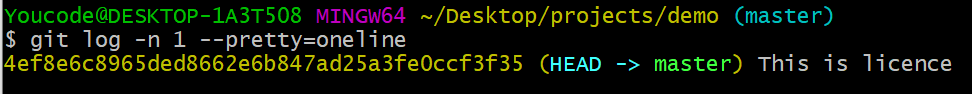
****

**6⇨ les fichiers traqués.**

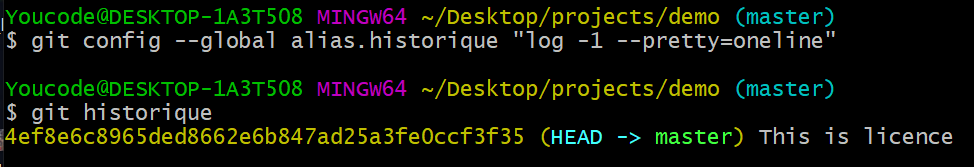
****

***STEP3***

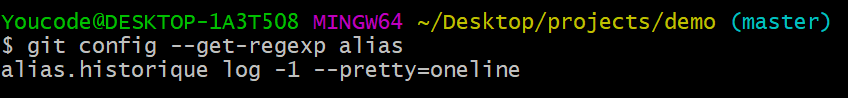
**1 : Affichez le dernier commit sur une ligne en ajoutant l’option d’affichage de la hiérarchie de la branche, avec les commit et leur branche aussi**

****

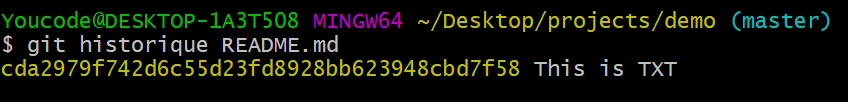
**2 : Créez un alias de la commande précédente le nom de l’alias est : historique**

****

**3 Affichez la liste des alias**

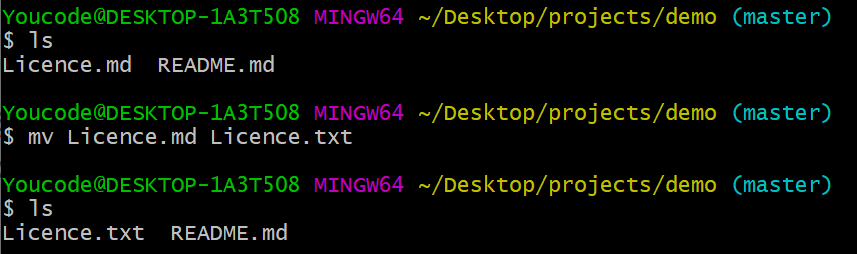
****

**4 Affichez l’historique des commit du fichier README.md avec l’alias.**

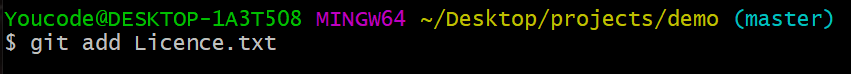
****

***Step 4***

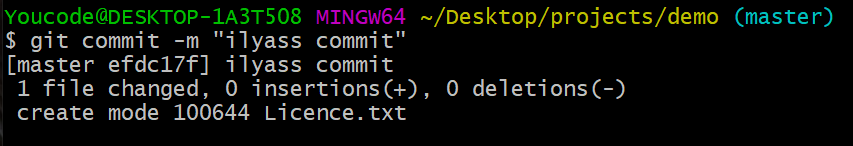
**⇨ Renommer le fichier Licence.md à Licence.txt**

****

**⇨ Faites le stagging avec mise à jour (ne pas faire « git add . »)**

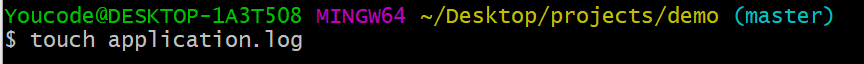
****

**⇨ Faites-le commit.**

****

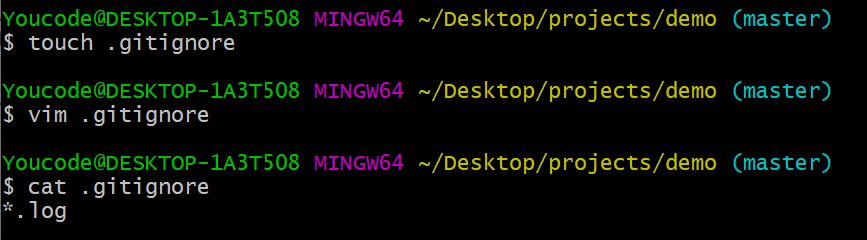
**⇨ Supposons que nous développons sur notre plateforme, surement on a des fichiers qu’on veut exclure de notre arborescence du repository Local**

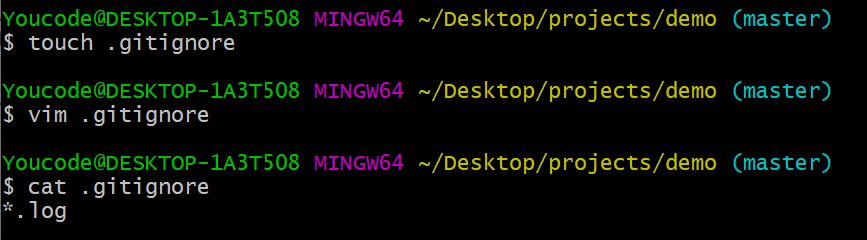
**1. Créez un fichier nommé application.log**

****

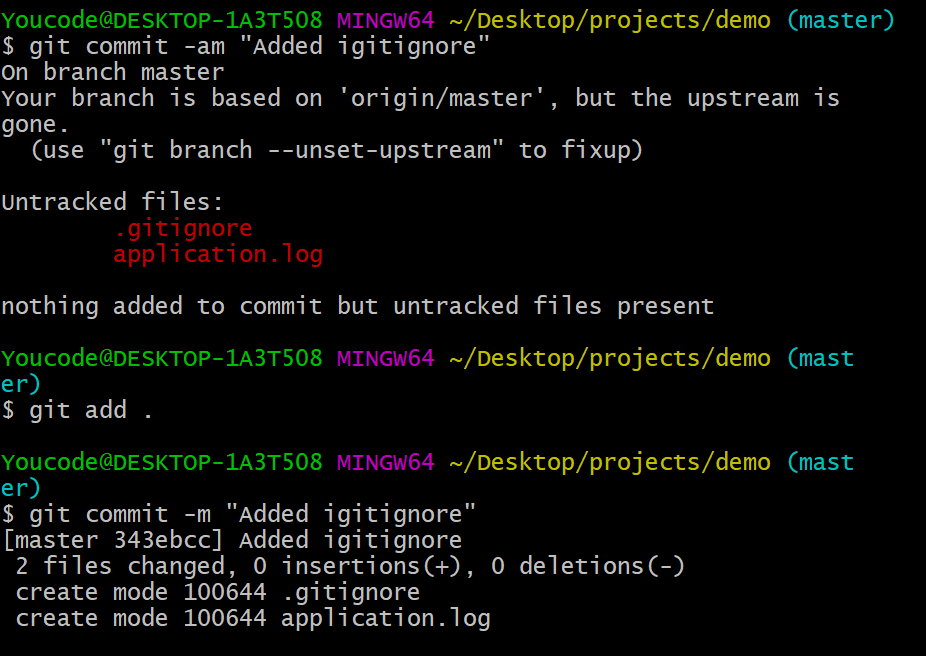
**2. Ne faites pas le stagging mais créez un fichier nommé**

**« .gitignore »**

****

**3. Sur le fichier .gitignore Ajoutez la ligne suivante « \*.log »**

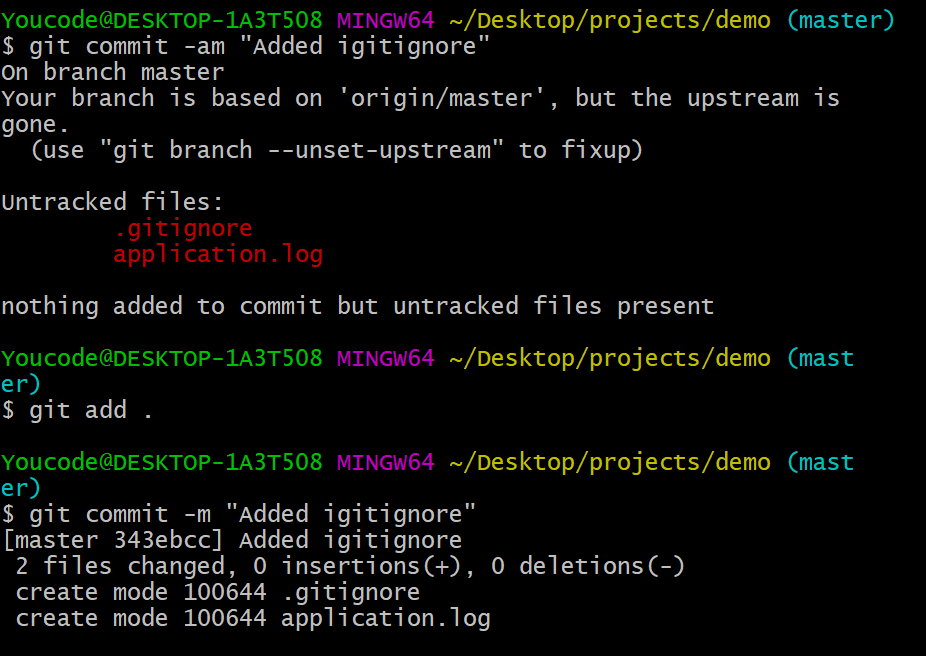
**4. Faites le stagging et le commit**

****

**5. Qu’est-ce que vous constatez ?**

Nous avons collecté “git add et git commit” dans la même ligne mais il on pas commité parce-que le fichier Il vient créez maintenant “.gitignore”

la solution on va séparer “git add” et “git commit”

****

***Step 5:***

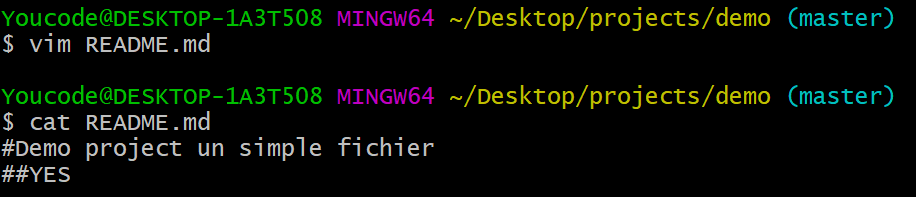
**Introduction : un petit rappel, on a trois types pour faire le merge :**

**⇨ The fast forward merging**

**⇨ The manual**

**⇨ The automatic**

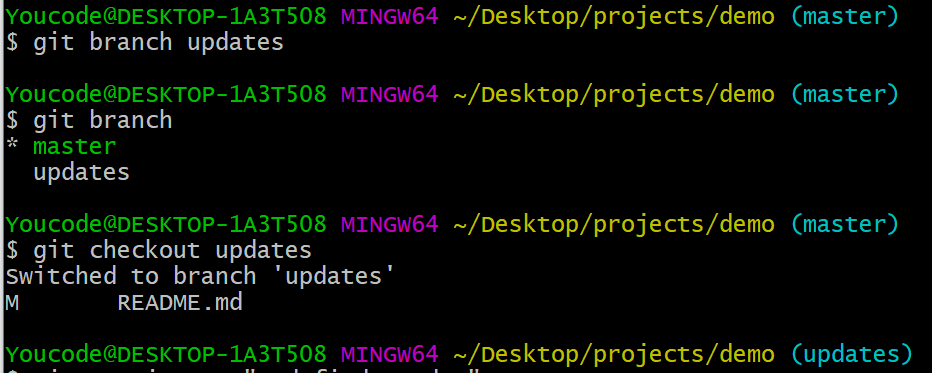
**1 Modifiez le fichier README.md**

****

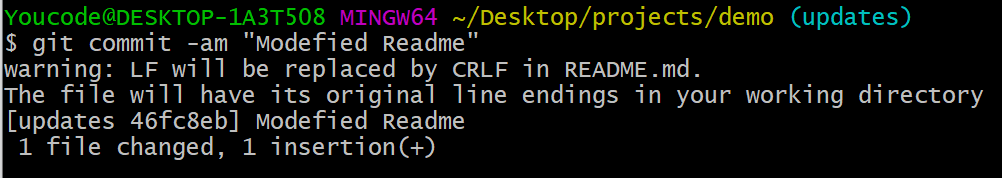
**2 Ne faites pas le commit**

**OK**

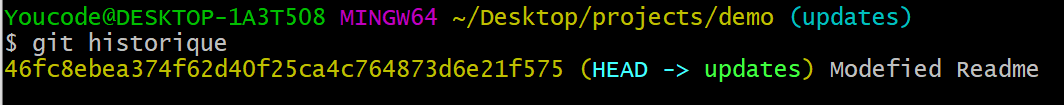
**3 Créez une branche pour la modification du fichier README.md du nom ‘updates’**

****

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

****

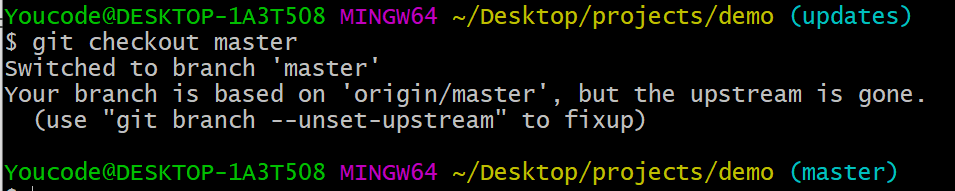
**5 Affichez l’historique avec l’alias**

****

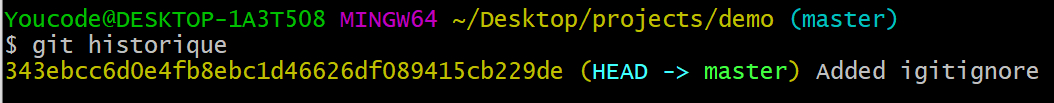
**6 Qu’est-ce que vous constatez ?**

on voit une nouvelle commit créez dans la branch updates

**7 Retournez vers la branche Master.**

****

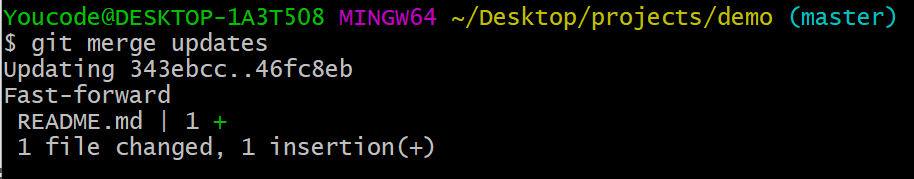
**8 Affichez l’historique avec l’alias**

****

**9 Qu’est-ce que vous constatez**

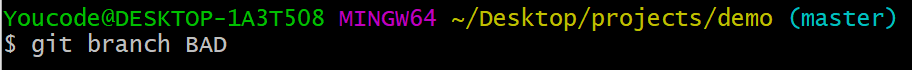
On voit que la dernier commit est affiché ou branch master , on conclure que chaque branch a une historique de commit .

**10 Maintenant faites le merge.**

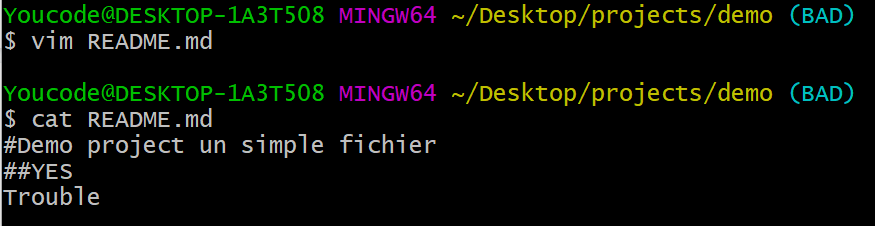
****

***Step 6:***

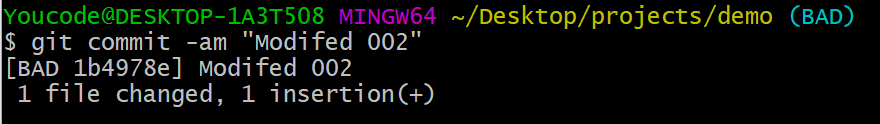
**1 Créez une branche avec le nom ‘BAD’**

****

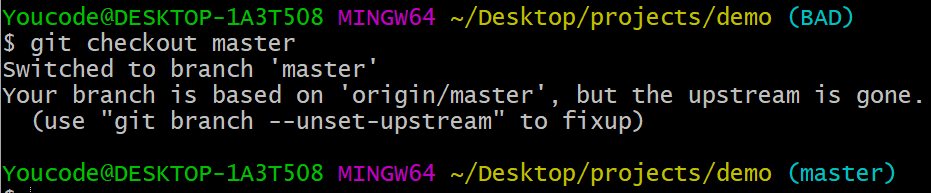
**2 Modifiez le fichier README.md et ajouter la ligne ‘Trouble’**

****

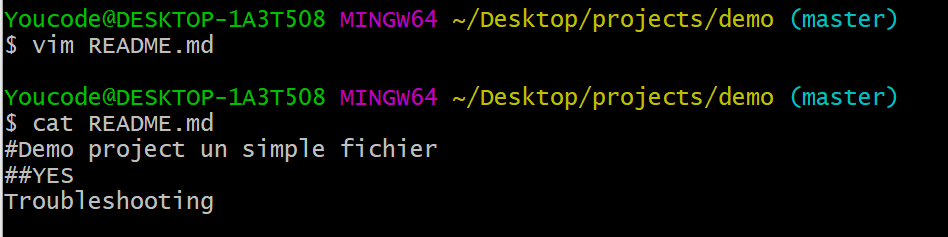
**3 Faites le stagging et le commit en une seule ligne**

****

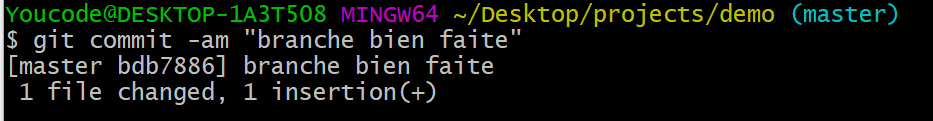
**4 Switchez faire la branche principale ‘master’**

****

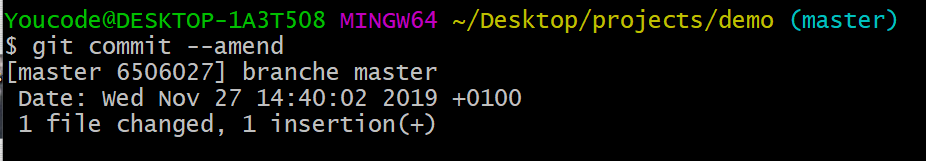
**5 Maintenant modifiez le fichier README.md et ajoutez la ligne ‘Troubleshooting’**

****

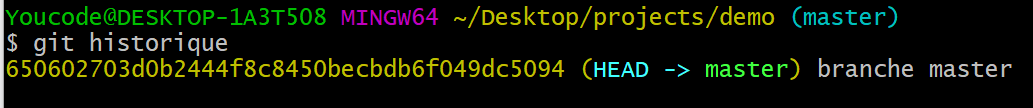
**6 Stagging/commiting avec commentaire ‘branche bien faite’**

****

**7 Oups !! corriger le commentaire du dernier commit**

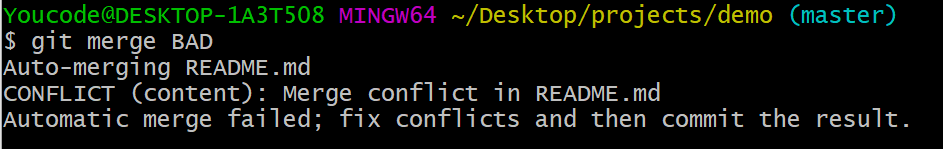
****

**8 Affichez l’historique**

****

**9 Qu’est-ce que vous constatez ? expliquez l’objectif de ce petit scenario**

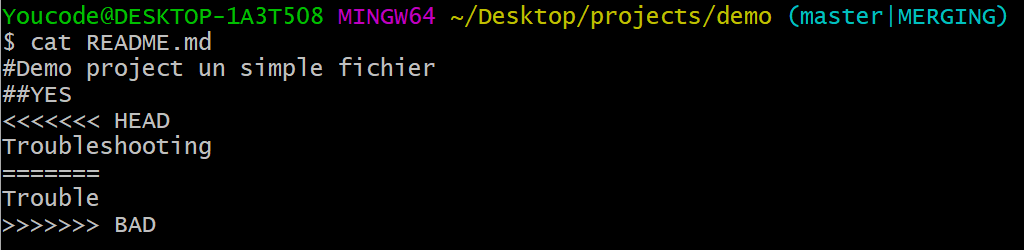
**10 Faites le merge de la branche ‘BAD’**

****

**11 Expliquez le message ?**

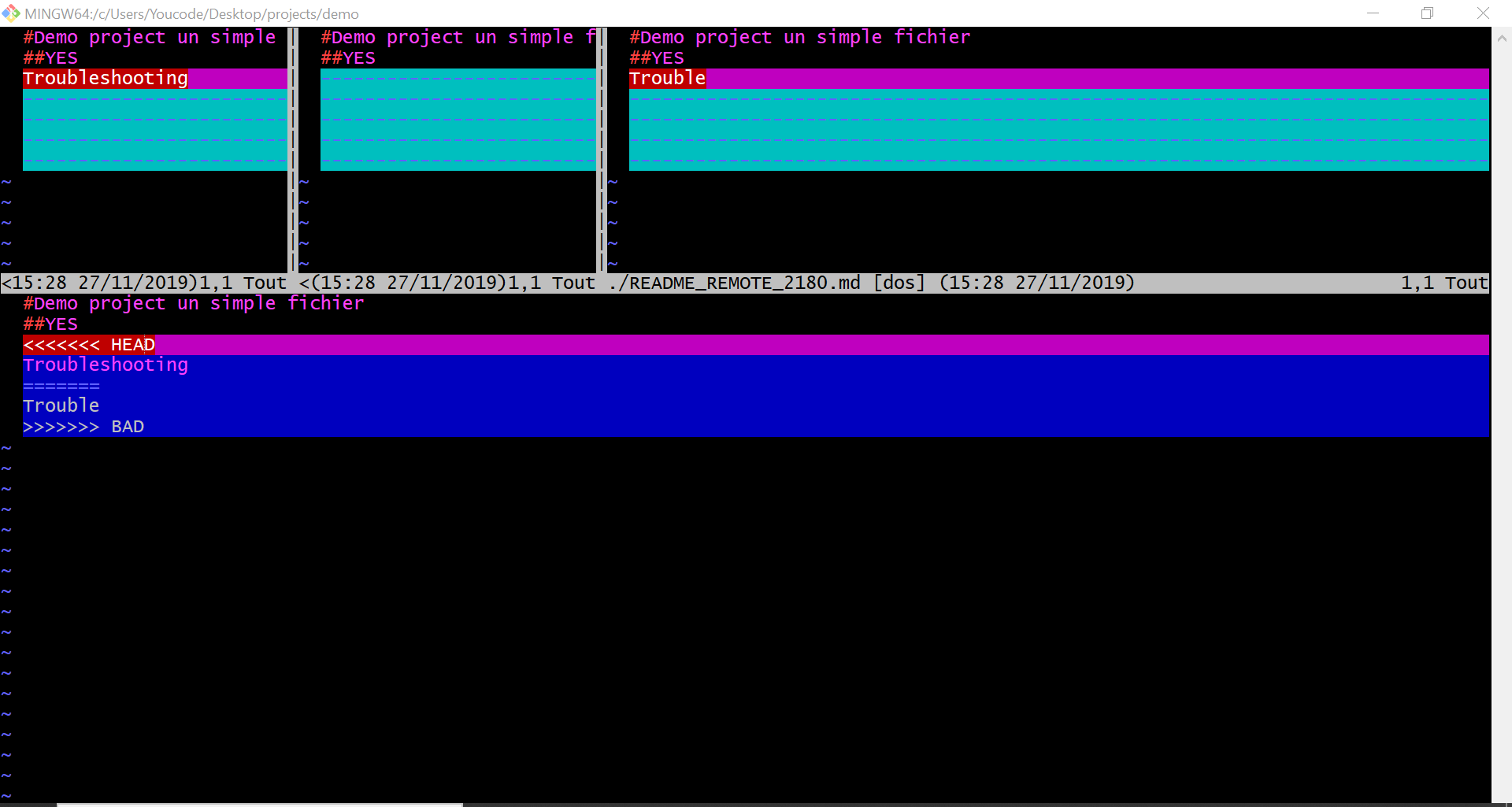
On a un problème aux merge , parce que on a modifié le fichier readme.md dans branch master et branch bad a la même ligne

**12 Exécutez la commande suivante : cat README.md ? Expliquez ?**

****

La commande cat il nous aider de connais la place de conflit , entre HEAD et BAD

**13 Tapez la commande « git mergetool » ? qu’est-ce que vous constatez ?**

****

**Quand n’a taper la commande “Git mergetool “ 4 éditeure VIM est afficher :**

1er folder: 1er modification

2éme folder : Avant la modification

3émé folder : 2éme modification

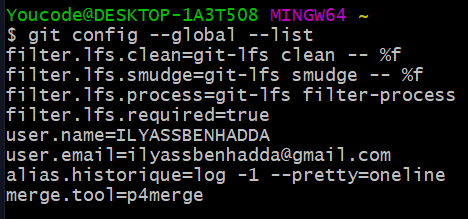
4éme folder : le conflit

***Step 7:***

1 Installez p4merge sur votre Pc

**ok**

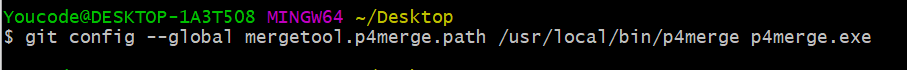
2 Sur git bash tapez la commande : git config –global –list



3 Git config –global merge.tool p4merge

**OK**

4 Git config –global mergetool.p4merge.path “lien d’installation” ( .exe)



5 Tapez la commande git mergetool

**ok**

6 Analysez ce que vous voyez ? et expliquez la plateforme ouverte en temps réel

***Step 8:***

⇨ Sur le fichier .gitignore ; écrivez une clause pour rejeter les fichiers indésirables et redondants

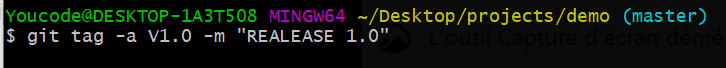
**Scénario 2**

***#1 : First Step : Tagging***

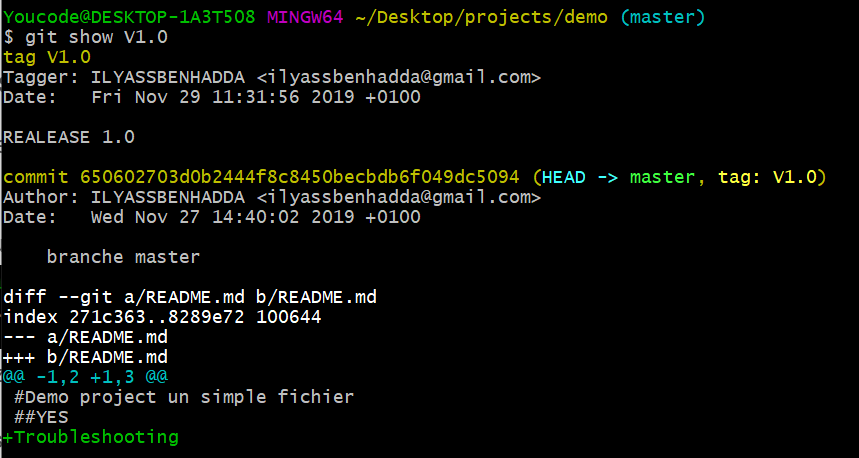
⇨ Déplacez-vous sur la branche Principale

**OK**

⇨ Créez un TAG avec un nom V1.0 et un commentaire ‘ REALEASE 1.0 ‘



⇨ Affichez les informations sur le TAG



⇨ D’après vous un **TAG** il sert à faire quoi au juste ?

**Git tag** sert à marquer des points spécifiques de l'historique d'un référentiel comme étant importants. Généralement, les utilisateurs utilisent cette fonctionnalité pour marquer les points de publication.

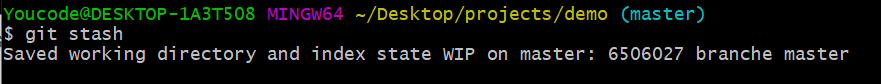
***#2 : Second Step : Stashing and Saving work in Progress***

⇨ Modifiez le fichier README.md , ajoutez une ligne

**OK**

⇨ A vrai dire , l’équipe a décidé que cette tâche sera en standby et que les modifications progressent avec le timeline du projet

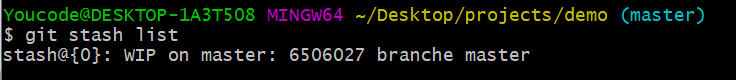
1. Tapez la commande git Stash ?



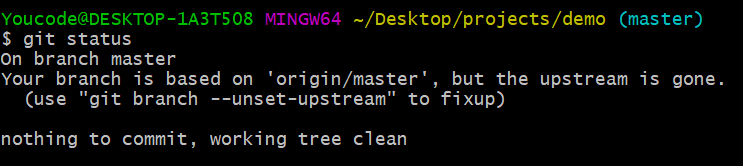
2. Expliquez le fonctionnement de la commande git Stash

Utilisez cette option git stash lorsque vous souhaitez enregistrer l'état actuel du répertoire de travail et de l'index, mais que vous souhaitez revenir à un répertoire de travail vide

3. Tapez la commande git stash list



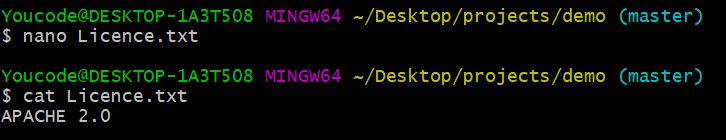
4. Exécutez la commande git status ?



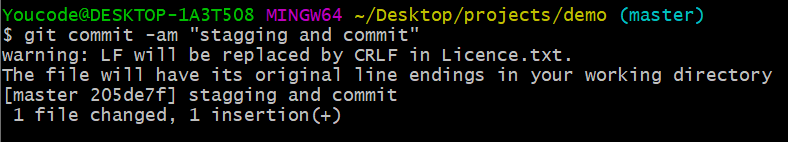
5. Qu’est-ce que vous constatez ?

Dans le branche master aucun commit été enregistré .

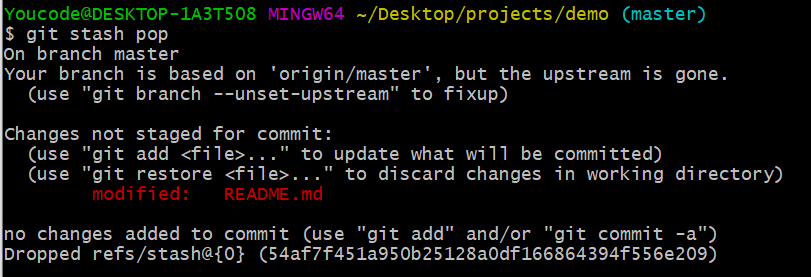
⇨ Modifiez le fichier « Licence.txt », ajoutez la ligne « APACHE 2.0 »



⇨ Faites stagging et le commit en une seule ligne



⇨ Exécutez la commande « git stash pop » ?



⇨ Qu’est-ce que vous constatez ?

Les données de stash seront supprimées de la pile de la liste de stash.git stash pop prend un changement caché, le supprime de la «pile cachée» et l'applique à votre arbre de travail actuel.

***#3: third step: Voyage sur Github, Local Repo to github Repo***

⇨ Pour cette phase vous devez avoir un compte GITHUB

ok

⇨ Créez un repo github public sans ajouter le fichier README.md

ok

⇨ Créez le remote en https

https://github.com/abdessamed11/scenario2.git

⇨ Examinez le remote

**Youcode@DESKTOP-KRPHUO1 MINGW64 ~/Desktop/projects (master)**

**$ git remote add origin https://github.com/abdessamed11/scenario2.git**

⇨ Pushez le tous à travers la commande : git push –u origin master - -tags

**Youcode@DESKTOP-KRPHUO1 MINGW64 ~/Desktop/projects (master)**

**$ git push -u origin master --tags**

✔ Expliquez la commande ?

**GIT PUSH Utilisation de la ligne de commande pour PUSH to GitHub**

**$ git push -u origin master --tags cette commande Met à jour les références distantes à l'aide de références locales en transfert même les tags créés.**

✔ Expliquez les options –u et - -tags ?

git push -u pour pusher le dossier modifier sur le dossier distant et fait la mise à jour

--tags pour pusher le dossier local avec les tags deja creer.

⇨ Sur Github Vérifiez la liste des commits , les branches , les releases et les tags



***#4 : Fourth Step : Mini challenge ( optionnel)***

En examinant les types d’authentification sur GITHUB, on tombe sur l’authentification HTTPS et SSH, certes HTTPS est beaucoup plus facile à manager tandis que le SSH est plus sécurisé tandis que fiable en ce qui concerne les transferts cryptés.

Le but de ce challenge est de créer une authentification SSH entre votre repo local et le repo GITHUB.

▪ HINTS :

🡺 Créez un fichier sous git nommé **.SSH**

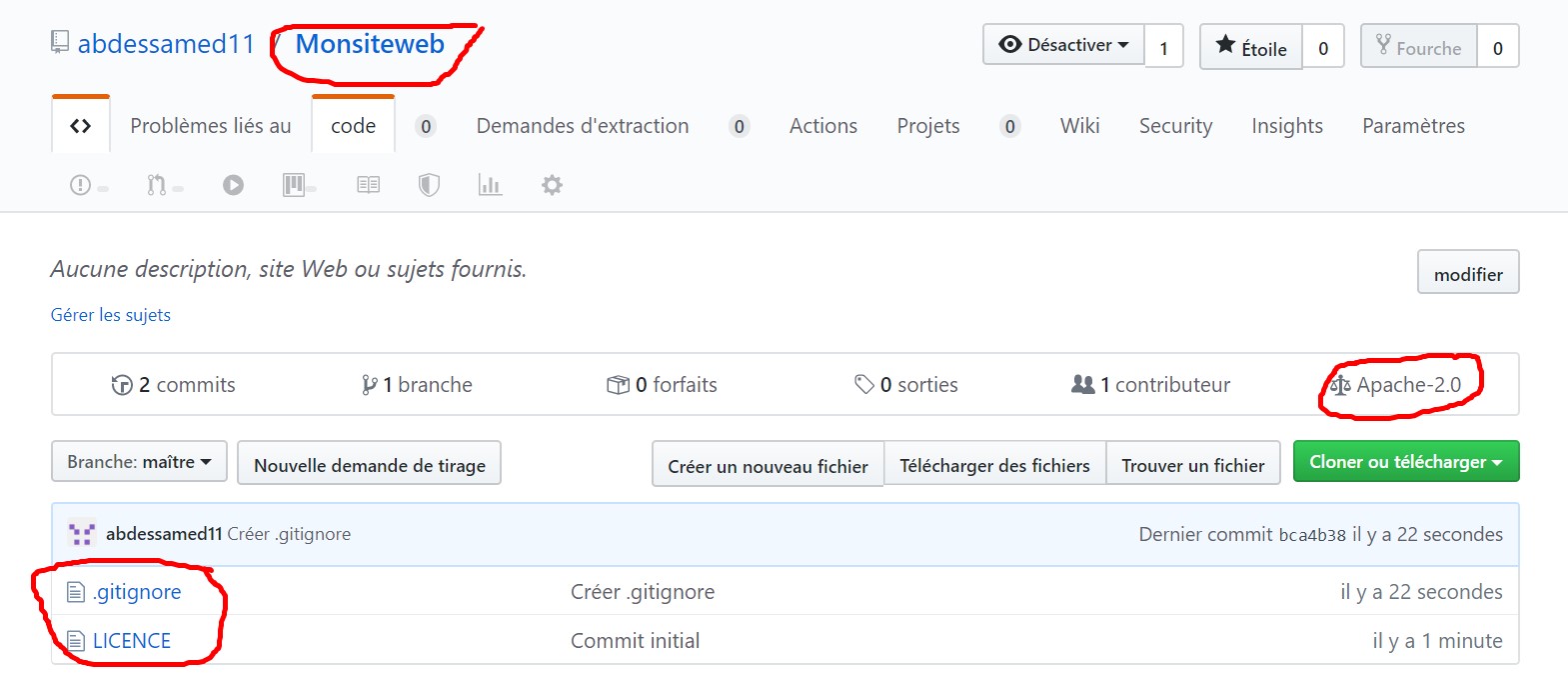
⇨ Déplacez-vous dans le fichier .SSH

⇨ Maintenant à vos mains : Créez une authentification sécurisé SSH entre votre repo local et votre repo distant (aide : Ssh –keygen-t rsa –C ' email’).

***#5: Fifth Step: Création d’une local copy :***

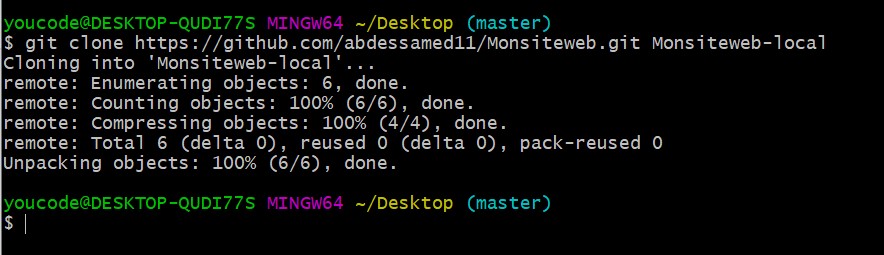
⇨ Sur github créez un autre repo nommé ( Monsiteweb)

⇨ Ajoutez à l’arboresence toujours sur github le fichier **.gitignore** et un fichier **licence.txt ‘APACHE 2.0 ‘**



⇨ Déplacez-vous dans le répertoire projets sous GIT

⇨ Créez un clone github vers le local sous le nom (Monsiteweb-local)



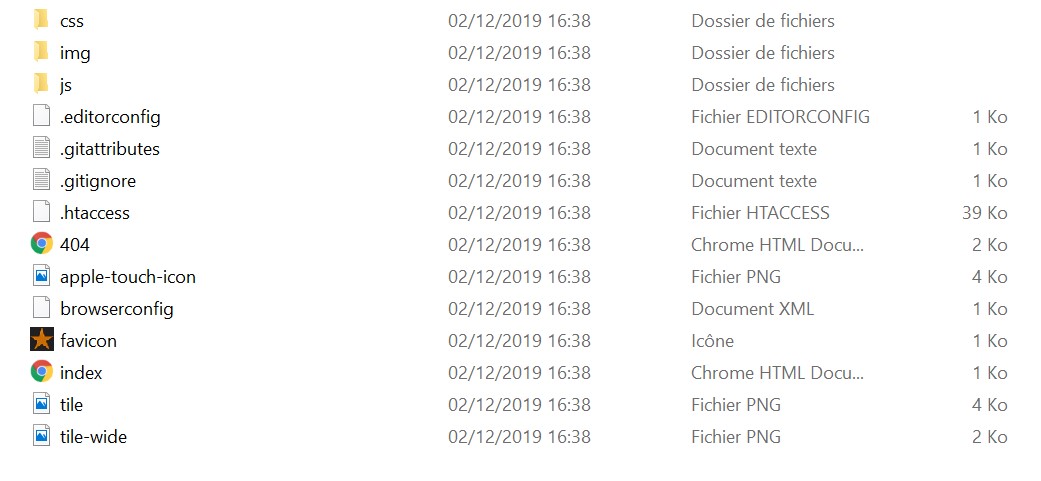
⇨ Vérifiez si le clone

***#6: sixth step : Sending the website :***

⇨ Télécharger le site web depuis le lien suivant :<http://www.initializr.com/>

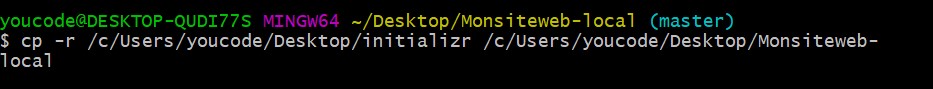
⇨ Sur le site telecharger un site bootstrap avec le fichier **.htaccess** et le fichier **404.html**

⇨ Analysez l’arboresence.

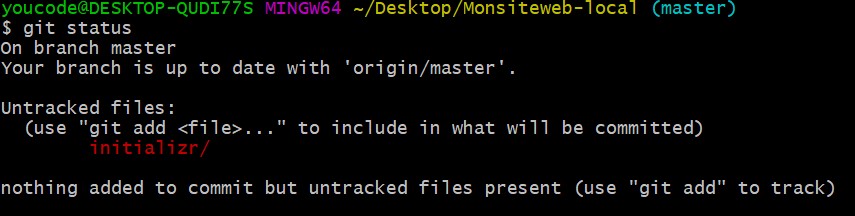
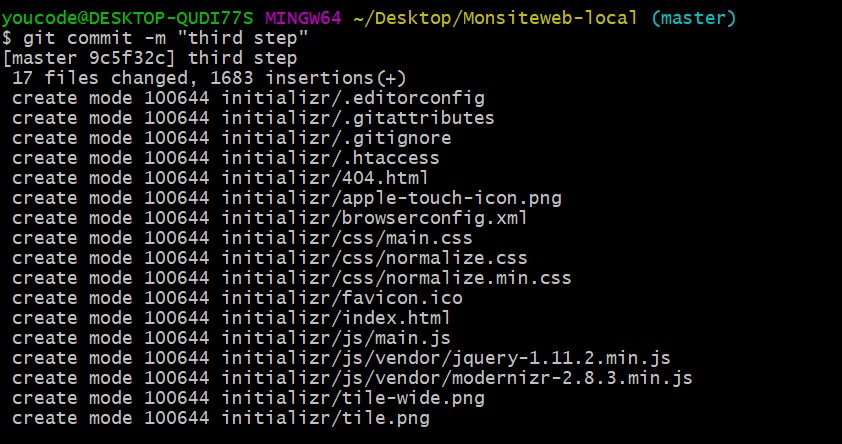


⇨ Expliquez ?

⇨ Copiez le site télécharger dans votre repo local à travers une seule commande :



⇨ Git status



⇨ Faites le stagging et le commit en une seule ligne

⇨ Faite le push à github



⇨ Vérifiez l’existence du site local sur votre repo github

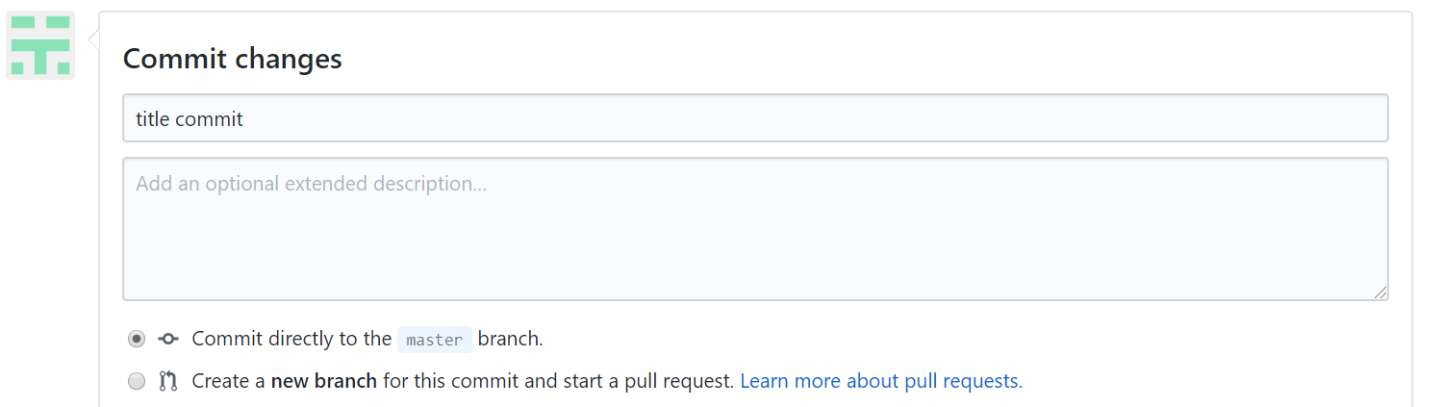
Ok

***#7: seventh step : Fetch and pull*** :

⇨ Sur github éditez le fichier **Index.html**, sur la balise <title> </title> ajoutez le titre , mon premier site web .

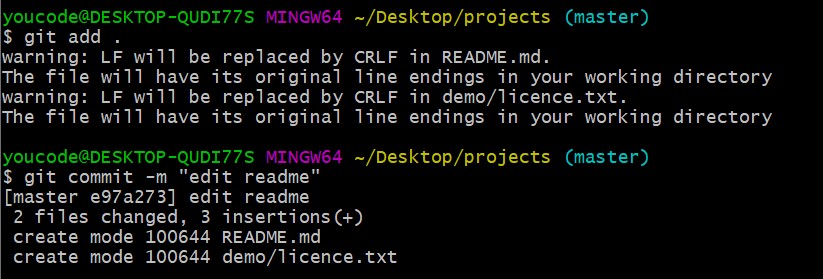


⇨ Faites le commit sur github



⇨ Sur git et sur le repo local , Editez le fichier README.md

⇨ Faites le stagging et le commit en une seule ligne

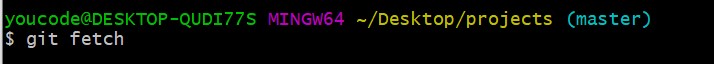


⇨ Qu’est-ce que vous constatez ?

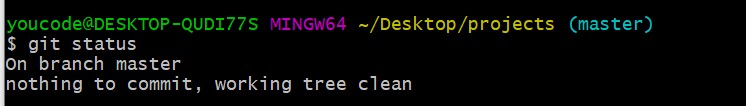
La commit est bien fait

● Trouble shooting :

⇨ Executez la commande « git fetch »



⇨ Git status



⇨ Expliquez ?

⇨ Git pull

**$ git pull origin master**

**From https://github.com/khaoulabendaoud/Monsiteweb**

**\* branch master -> FETCH\_HEAD**

**Merge made by the 'recursive' strategy.**

**initializr/index.html | 2 +-**

**1 file changed, 1 insertion(+), 1 deletion(-)**

⇨ Git push

**$ git push origin master**

**Enumerating objects: 9, done.**

**Counting objects: 100% (8/8), done.**

**Delta compression using up to 8 threads**

**Compressing objects: 100% (4/4), done.**

**Writing objects: 100% (5/5), 541 bytes | 541.00 KiB/s, done.**

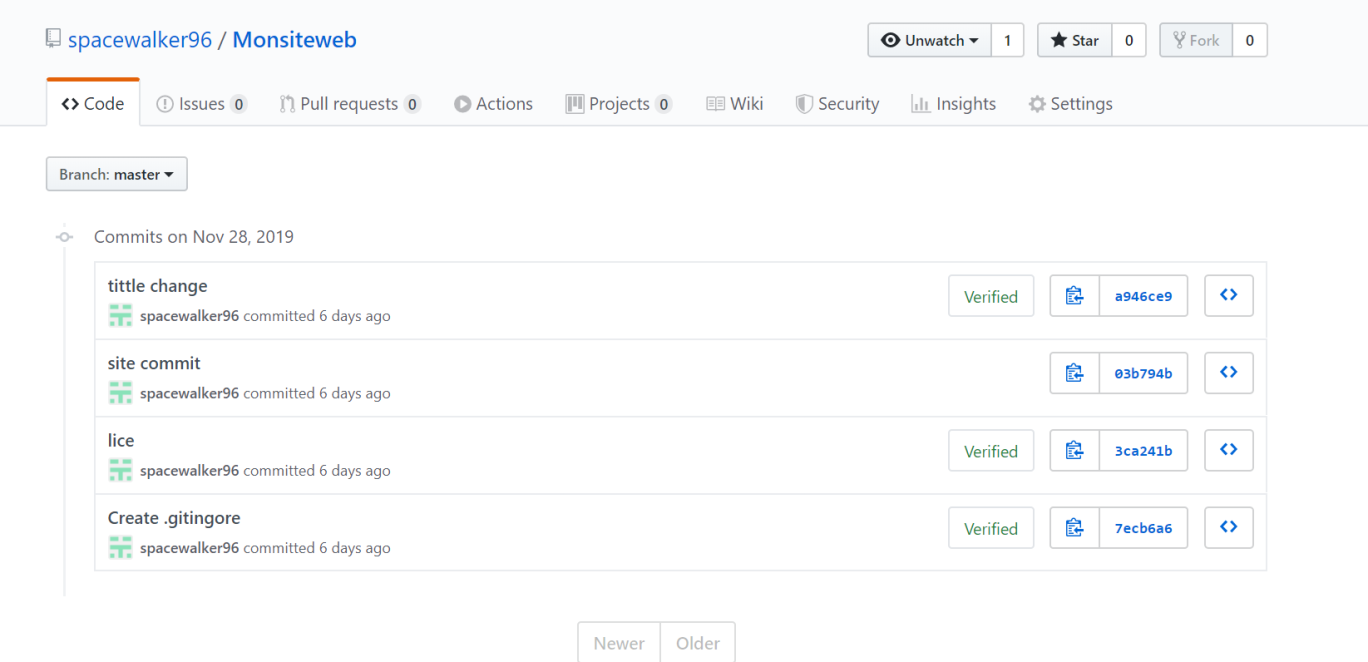
**Total 5 (delta 2), reused 0 (delta 0)**

**remote: Resolving deltas: 100% (2/2), completed with 1 local object.**

**To https://github.com/khaoulabendaoud/Monsiteweb.git**

**8a526e4..c3fab08 master -> master**

⇨ Verifiez les commit sur github



***Scénario 3***

Brief Scenario 3

***#1: First Step: Changes on Github***

Sur votre compte Github : faites les modifications suivantes :

1. Sur le dossier CSS ajoutez un autre fichier nommée style.css

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

récente </h1>

3. 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 commits .git

2. Sur GITBASH, vérifiez avec « git status »

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

3. Visualisez et télécharger les fichiers distants sur GITBASH

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git fetch

remote: Enumerating objects: 7, done.

remote: Counting objects: 100% (7/7), done.

remote: Compressing objects: 100% (5/5), done.

remote: Total 5 (delta 3), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (5/5), done.

From https://github.com/adamkhairi/Monsiteweb

9a6fa78..70c1d9f master -> origin/master

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

$ git pull

Updating 9a6fa78..70c1d9f

Fast-forward

404.html => error404.html | 0

index.html | 1 +

2 files changed, 1 insertion(+)

rename 404.html => error404.html (100%)

5. Vérifiez votre repo Local

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ ls -la

total 55

drwxr-xr-x 1 Youcode 197121 0 Dec 1 00:13 **.**/

drwxr-xr-x 1 Youcode 197121 0 Nov 30 23:51 **..**/

drwxr-xr-x 1 Youcode 197121 0 Dec 1 00:13 **.git**/

-rw-r--r-- 1 Youcode 197121 2 Nov 30 23:51 .gitignore

-rw-r--r-- 1 Youcode 197121 3959 Nov 30 23:51 apple-touch-icon.png

-rw-r--r-- 1 Youcode 197121 428 Nov 30 23:51 browserconfig.xml

drwxr-xr-x 1 Youcode 197121 0 Nov 30 23:51 **css**/

-rw-r--r-- 1 Youcode 197121 1332 Dec 1 00:13 error404.html

-rw-r--r-- 1 Youcode 197121 766 Nov 30 23:51 favicon.ico

drwxr-xr-x 1 Youcode 197121 0 Nov 30 23:51 **fonts**/

-rw-r--r-- 1 Youcode 197121 5451 Dec 1 00:13 index.html

drwxr-xr-x 1 Youcode 197121 0 Nov 30 23:51 **js**/

-rw-r--r-- 1 Youcode 197121 11558 Nov 30 23:51 LICENSE.txt

-rw-r--r-- 1 Youcode 197121 29 Nov 30 23:51 README.md

-rw-r--r-- 1 Youcode 197121 3482 Nov 30 23:51 tile.png

-rw-r--r-- 1 Youcode 197121 1854 Nov 30 23:51 tile-wide.png

6. Sur votre repo Distant GITHUB, laissez un commentaire sur les fichiers que vous avez modifiez

en regardant la liste des commits.

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git log index.html

commit d29e3475f2d46ae35385d9cf9028a5a4e4847768

Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

Date: Sat Nov 30 23:57:16 2019 +0100

add H1

commit 33535a8be87b1e2119084c70976bbbfafe4e8415

Author: ILYASS BENHADDA <57941607+ilyassbenhadda@users.noreply.github.com>

Date: Wed Nov 27 21:09:17 2019 +0100

Update title

commit ff069e07f42a3ab0723177bbce800ab49235f49e

Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

Date: Wed Nov 27 21:00:15 2019 +0100

copy intializr

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

*1. Repo distant :*

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

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

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

Compare and pull request : To create a pull request and merge the branch with master

*2. Repo local :*

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch annulation

*Sur votre repo local éditez le fichier « index.html » et supprimez la balise H1.*

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (annulation)

$ git commit -am 'delete h1'

[annulation 12526a5] delete h1

1 file changed, 1 deletion(-)

*« Git show » pour vérifier vos changements.*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (annulation)

$ git show

commit 12526a5993ba1c007b2658956f5d8019bc8cb3cc (**HEAD -> annulation**)

Author: ILYAS BENHADDA <ilyassbenhadda@gmail.com>

Date: Sun Dec 1 00:53:47 2019 +0100

delete h1

**diff --git a/index.html b/index.html**

**index b804ec9..8db9e35 100644**

**--- a/index.html**

**+++ b/index.html**

@@ -24,7 +24,6 @@

<script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></script>

</head>

<body>

- <h1> modification récente </h1>

<!--[if lt IE 8]>

<p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="http://browsehappy.com/

">upgrade your browser</a> to improve your experience.</p>

<![endif]-->

*3. Vérification :*

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (annulation)

$ git push origin annulation

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.

Total 3 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

remote:

remote: Create a pull request for 'annulation' on GitHub by visiting:

remote: https://github.com/adamkhairi/Monsiteweb/pull/new/annulation

remote:

To https://github.com/adamkhairi/Monsiteweb.git

\* [new branch] annulation -> annulation

*Maintenant vérifiez votre repo distant.*

*Qu’est-ce que vous constatez.*

***#3: Third Step: 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 ?*

Apres la modification on delivre une pull request au chef ou master du projet pour la validation de la modification

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

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

Après la création de la requête de pull et une analyse du requête, On a page avec des informations sur la modification faites au niveau de Branch et le system propose de faire le merge du branch annulation avec la branch de base

*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.*

***#4: fourth Step: 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 :*

*body {*

*Width : 100% ;*

*Height : 100%*

*}*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch ajustement

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git checkout ajustement

Switched to branch 'ajustement'

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (ajustement)

$ code css/main.css

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (ajustement)

$ git commit -am 'add body css'

[ajustement 2f16777] add body css

1 file changed, 5 insertions(+), 1 deletion(-)

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (ajustement)

$ git push origin ajustement

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 383 bytes | 383.00 KiB/s, done.

Total 4 (delta 3), reused 0 (delta 0)

remote: Resolving deltas: 100% (3/3), completed with 3 local objects.

To https://github.com/adamkhairi/Monsiteweb.git

70c1d9f..2f16777 ajustement -> ajustement

*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’*

*2. Un pull Request est demandé sur GITHUB ! N’oubliez pas !*

*3. Sur GITBASH, revenez vers la branche master ! Expliquez le rendu de la commande?*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (ajustement)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

Un message qui dit que la Branch et est à jour avec la branch origin/Master

*4. Maintenant demandez le pull ?*

*5. Une fois la commande du pull est exécutée, à votre avis quel type de merge on aura besoin*

*pour faire un merge sans conflits*

*6. Faites le merge à travers GITBASH*

*7. Une fenêtre s’ouvre après l’exécution de la commande du merge ; ce qui signifie le message*

*délivré ? à quoi sert cette fenêtre ?*

C’est une zone pour expliquer la cause du merge

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git merge ajustement

Merge made by the 'recursive' strategy.

css/main.css | 6 +++++-

1 file changed, 5 insertions(+), 1 deletion(-)

*8. Git status*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git status

On branch master

Your branch is ahead of 'origin/master' by 2 commits.

(use "git push" to publish your local commits)

Untracked files:

(use "git add <file>..." to include in what will be committed)

"sd\357\200\233OB"

nothing added to commit but untracked files present (use "git add" to track)

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git add .

warning: LF will be replaced by CRLF in sdOB.

The file will have its original line endings in your working directory

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git push origin master

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (2/2), 297 bytes | 297.00 KiB/s, done.

Total 2 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/adamkhairi/Monsiteweb.git

a90fc79..6ac93fc master -> master

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

*changements ?*

*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 ?*

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

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

Les changements sont pas encore effectué

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch -a

ajustement

annulation

\* master

remotes/origin/Example

remotes/origin/HEAD -> origin/master

remotes/origin/add-license-1

remotes/origin/ajustement

remotes/origin/annulation

remotes/origin/master

*4. 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 ?*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git pull origin master

From https://github.com/ilyassbenhadda/Monsiteweb

\* branch master -> FETCH\_HEAD

Already up to date.

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

Branch wasn’t deleted

***#5: fifth Step: 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 ;*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git pull --all

Fetching origin

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/adamkhairi/Monsiteweb

\* [new branch] updatelicence -> origin/updatelicence

Already up to date.

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git merge origin/updatelicence

Updating 6ac93fc..2eba829

Fast-forward

LICENSE.txt | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

*Executez Git push !*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git push

Total 0 (delta 0), reused 0 (delta 0)

To https://github.com/adamkhairi/Monsiteweb.git

6ac93fc..2eba829 master -> master

*Git branch –d ‘updatelicence’*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch -d origin/updatelicence

error: branch 'origin/updatelicence' not found.

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch -a

ajustement

annulation

\* master

remotes/origin/Example

remotes/origin/HEAD -> origin/master

remotes/origin/add-license-1

remotes/origin/ajustement

remotes/origin/annulation

remotes/origin/master

remotes/origin/updatelicence

*Maintenant sur GITBASH, trouvez la commande qui va nous permettre de supprimer la branche depuis le remote et ainsi de pushez les changements de la suppression simultanément ?*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git push origin --delete updatelicence

To https://github.com/adamkhairi/Monsiteweb.git

- [deleted] updatelicence

*Vérifiez vos Repo (local et distant)*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git branch -a

ajustement

annulation

\* master

remotes/origin/Example

remotes/origin/HEAD -> origin/master

remotes/origin/add-license-1

remotes/origin/ajustement

remotes/origin/annulation

remotes/origin/master

*Pourriez-vous expliquez l’objectif de cette phase ?*

L’objectif de cette phase est d’expliqué l’utilité des branchs

***#6: sixth step: Rebasing:***

*Introduction générale : Rebase vs merge*

*Faites une veille sur REBASE de 30 min intergroupe , visitez le lien suivant :*

*https://www.grafikart.fr/tutoriels/amend-rebase-588*

*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 suivante*

*Merged*

*feature A*

*Master*

*Feature A*

*Quand faire un merge ?*

*Sur des branches durables*

*Pour indiquer un évènement*

*marquant : Release, Sprint etc.*

*Merged*

*feature A*

*Master*

*Feature A*

*Quand faire rebase ?*

*Sur acceptation d’un pull Request*

*d’un feature ou suivi d’un merge*

*Lors d’un git pull*

*Master*

*Feature A*

*j*

*Feature A*

*<!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-*

*ie7" lang=""> <![endif]-->*

*<!--[if IE 7]> <html class="no-js lt-ie9 lt-*

*ie8" lang=""> <![endif]-->*

*<!--[if IE 8]> <html class="no-js lt-ie9" lang=""> <![endif]-->*

*<!--[if gt IE 8]><!-->*

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

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ code index.html

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git commit -am 'changing title'

[master dcf83dd] changing title

2 files changed, 70 insertions(+), 4 deletions(-)

create mode 100644 "sd\357\200\233OB"

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/adamkhairi/Monsiteweb

2eba829..c66cc30 master -> origin/master

*Git status ? lisez les commentaires*

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git status

On branch master

Your branch and 'origin/master' have diverged,

and have 1 and 1 different commits each, respectively.

(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git pull --rebase

First, rewinding head to replay your work on top of it...

Applying: changing title

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

Youcode@SpectRum MINGW64 ~/Desktop/Projects2/Monsiteweb-local (master)

$ git historique

\* commit 273a931422fdbb17d49b0eb43d4fafd78f24405b (**HEAD -> master**)

| Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

| Date: Sun Dec 1 02:23:27 2019 +0100

|

| changing title

|

\* commit c66cc30c2a5316dff83874bd1b0c2380795f79d6 (**origin/master**, **origin/HEAD**)

| Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 02:16:05 2019 +0100

|

| add updates line

|

\* commit 2eba829e1510857e9095ebb9b49e5be2cdcadab4

| Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 01:54:34 2019 +0100

|

| apache 3.0

|

\* commit 6ac93fccdc284852dd50f6d6e2c668246bc3b19d

|\ Merge: a90fc79 2f16777

| | Author: ILYASS BENHADDA <ilyassbenhaddda@gmail.com>

| | Date: Sun Dec 1 01:28:10 2019 +0100

| |

| | Merge branch 'ajustement'

| | 4 - 7 Merge branch ajust

| |

| \* commit 2f1677756c3fd23c99ed8e73fcf950e5cd4ba33b (**origin/ajustement**, **ajustement**)

| | Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

| | Date: Sun Dec 1 01:20:36 2019 +0100

| |

| | add body css

| |

\* | commit a90fc795f74e0cab223c9e52e41fcdb92e573158

|\ \ Merge: 70c1d9f 12526a5

| |/ Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

|/| Date: Sun Dec 1 01:16:47 2019 +0100

| |

| | Merge pull request #2 from adamkhairi/annulation

| |

:...skipping...

\* commit 273a931422fdbb17d49b0eb43d4fafd78f24405b (**HEAD -> master**)

| Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

| Date: Sun Dec 1 02:23:27 2019 +0100

|

| changing title

|

\* commit c66cc30c2a5316dff83874bd1b0c2380795f79d6 (**origin/master**, **origin/HEAD**)

| Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 02:16:05 2019 +0100

|

| add updates line

|

\* commit 2eba829e1510857e9095ebb9b49e5be2cdcadab4

| Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 01:54:34 2019 +0100

|

| apache 3.0

|

\* commit 6ac93fccdc284852dd50f6d6e2c668246bc3b19d

|\ Merge: a90fc79 2f16777

| | Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

| | Date: Sun Dec 1 01:28:10 2019 +0100

| |

| | Merge branch 'ajustement'

| | 4 - 7 Merge branch ajust

| |

| \* commit 2f1677756c3fd23c99ed8e73fcf950e5cd4ba33b (**origin/ajustement**, **ajustement**)

| | Author: ILYASS BENHADDA <ilyassbenhada@gmail.com>

| | Date: Sun Dec 1 01:20:36 2019 +0100

| |

| | add body css

| |

\* | commit a90fc795f74e0cab223c9e52e41fcdb92e573158

|\ \ Merge: 70c1d9f 12526a5

| |/ Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

|/| Date: Sun Dec 1 01:16:47 2019 +0100

| |

| | Merge pull request #2 from adamkhairi/annulation

| |

| | delete h1

| |

| \* commit 12526a5993ba1c007b2658956f5d8019bc8cb3cc (**origin/annulation**, **annulation**)

|/ Author: ILYASSBENHADDA <ilyassbenhadda@gmail.com>

| Date: Sun Dec 1 00:53:47 2019 +0100

|

| delete h1

|

| \* commit 8ccdc5861245ad1745b97c210612d858870404f2 (**origin/Example**)

|/ Author: ILYASS BENHADDA<57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 00:21:08 2019 +0100

|

| 3th modification

|

\* commit 70c1d9f043e1d75ae7ee41df815dc320561f4bc2

| Author: ILYASS BENHADDA <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sun Dec 1 00:10:31 2019 +0100

|

| Rename 404.html to error404.html

|

\* commit d29e3475f2d46ae35385d9cf9028a5a4e4847768

| Author: adamkhairi <57996941+ilyassbenhadda@users.noreply.github.com>

| Date: Sat Nov 30 23:57:16 2019 +0100

|

| add H1

|

\* commit 9a6fa780f645dd2f5f1da4fbb37ef285a873e1a9

**|\** Merge: bff59e1 d654e3c

**|** **|** Author: adamkhairi <57996941+ilyassbenhadda@users.noreply.github.com>

**|** **|** Date: Sat Nov 30 23:49:50 2019 +0100

**|** **|**

**|** **|** Merge pull request #1 from adamkhairi/add-license-1

**|** **|**

**|** **|** Create LICENSE.txt

**|** **|**

**|** \* commit d654e3c01d5ce682294d7401bb0a2a3a1e228f0e (**origin/add-license-1**)

**|** **|** Author: ILYASS BENHADDA <57941607+DEVmajdaF@users.noreply.github.com>

**|** **|** Date: Wed Nov 27 16:46:20 2019 +0100

**|** **|**

**|** **|** Create LICENSE.txt

**|** **|**

\* **|** commit bff59e13e6433f5ca69a36daba473dc52d5c6c07

**|** **|** Author: ILYASS BENHADDA <57941607+DEVmajdaF@users.noreply.github.com>

**|** **|** Date: Fri Nov 29 11:05:14 2019 +0100

**|** **|**

**|** **|** Create Style.css

**|** **|**

\* **|** commit 4a29a9dc9330f8447fcb154b61590ff304ea208f

**|\** **\** Merge: 5795658 33535a8

**|** **|** **|** Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

**|** **|** **|** Date: Wed Nov 27 21:26:40 2019 +0100

**|** **|** **|**

**|** **|** **|** Message1 Merge branch 'master' of https://github.com/ILYASSBENHADDA/Monsiteweb

**|** **|** **|**

**|** \* **|** commit 33535a8be87b1e2119084c70976bbbfafe4e8415

**|** **|** **|** Author: ILYASS BENHADDA <57941607+ILYASSBENHADDA@users.noreply.github.com>

**|** **|** **|** Date: Wed Nov 27 21:09:17 2019 +0100

**|** **|** **|**

**|** **|** **|** Update title

**|** **|** **|**

\* **|** **|** commit 57956585e7f1d0cfc5b0597b967a5d583cfeffa5

**|/** **/** Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

**|** **|** Date: Wed Nov 27 21:13:21 2019 +0100

**|** **|**

**|** **|** editing Readme.md

**|** **|**

\* **|** commit ff069e07f42a3ab0723177bbce800ab49235f49e

**|/** Author: ILYASS BENHADDA <ilyassbenhadda@gmail.com>

**|** Date: Wed Nov 27 21:00:15 2019 +0100

**|**

**|** copy intializr

**|**

\* commit 9f635748a0a07b0a1fbc0879c4caefd1399b5b74

**|** Author: ILYASS BENHADDA <57941607+DEVmajdaF@users.noreply.github.com>

**|** Date: Wed Nov 27 16:41:26 2019 +0100

**|**

**|** Create .gitignore

**|**

\* commit 687558cad1f077ce4667922951a4b7350b6170b6

Author: ILYASS BENHADDA <57941607+DEVmajdaF@users.noreply.github.com>

Date: Wed Nov 27 16:39:50 2019 +0100

Initial commit

(END)

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

*Expliquez l’utilité du REBASE ?*

***#7: seventh step GitHub Insights:***

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

*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) .*

***#8 : eigtht step : 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 ?*

*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 : ..................*