RAPORT D'ACTIVITER

Ayoub essafi

Scénario #1

Objectifs :

- La formation d'un groupe de 5 apprenants
- Planification des taches selon une méthode de gestion de projet (SCRUM).
- La finalisation des Scénario dans les délais.
- La rédaction d'un rapport d'activités du groupe.
- La manipulation des commandes GIT.
- La découverte du monde GITHUB
- La gestion du temps, du stress et la finalisation des livrables.
- Les réunions de fin de Scénario
- La formalisation

First Step : Immersion

_ **Git** : Git est un système open source de gestion de versions distribué. Il a été conçu par Linux Torvald, le créateur du noyau Linux, pour gérer les sources de ce noyau que nous aimons tant. Git a été donc pensé dès le départ pour être efficace et rapide.

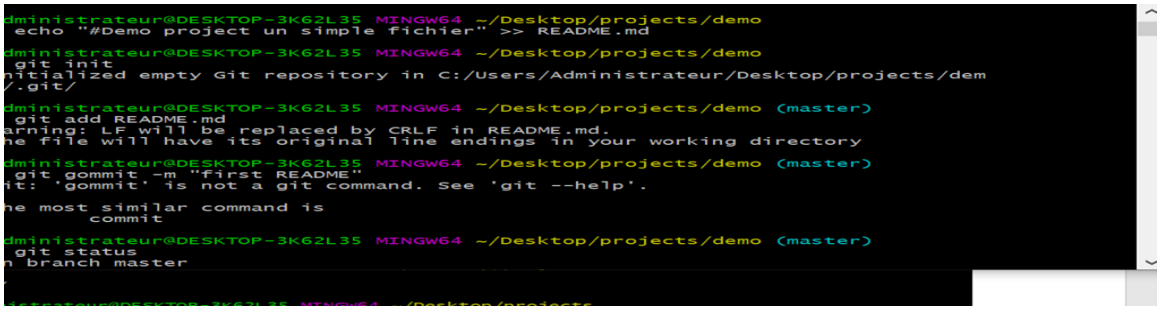
_ Le dossier **.git** contient toutes les informations nécessaires à votre projet dans le contrôle de version, ainsi que toutes les informations sur les validations, l'adresse du référentiel distant, etc. Toutes sont présentes dans ce dossier. Il contient également un journal qui stocke votre historique de validation afin que vous puissiez revenir à l'historique

_ créez un répertoire nommée « **/projects** » sur bureau

_ Créez un Repository Local

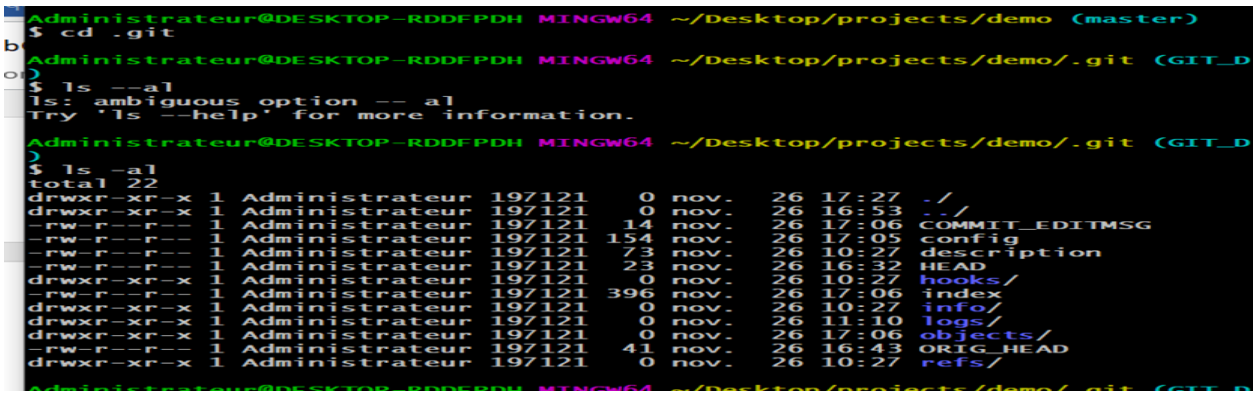
```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects
$ mkdir demo

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects
$ ls
demo/

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects
$ cd demo
```

_ Créez le fichier README.md et ajouter la ligne suivante : '#Demo project un simple fichier'.Faites le stagging et le commiting avec un commentaire.



2 : Second Step : La découverte

_ configuration « .**git** »



_Tapez Ls –al

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ ls -al
total 5
drwxr-xr-x 1 Administrateur 197121  0 nov.  26 11:01 ./
drwxr-xr-x 1 Administrateur 197121  0 nov.  26 10:59 ../
drwxr-xr-x 1 Administrateur 197121  0 nov.  26 11:04 .git/
-rw-r--r-- 1 Administrateur 197121 32 nov.  26 11:01 README.md
```

Les branches permettent d'avoir simultanément plusieurs versions de votre programme dans votre dépôt Git. C'est très utile, par exemple pour développer

une nouvelle fonctionnalité, tout en gardant la branche principale intacte. Ainsi, vous pouvez toujours faire des changements dans la branche principale (corrections de bugs par exemple), tout en développant en parallèle une nouvelle fonctionnalité.

HEAD est une référence au dernier commit de la branche de check-out en cours

Log : git log (affiche les commits log effectuées dans un projet)

_ créez le fichier « **Licence.md** ». et Faites-le Commit

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ touch Licence.md

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Licence.md

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

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git add Licence.md

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "add licence"
[master f171f6e] add licence
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Licence.md
```

_ Affichez les fichiers traqués.

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ ls
Licence.md  README.md
```

3 : third step : Historique

_ 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

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git log --oneline --graph --decorate --all
* f171f6e (HEAD -> master) add licence
* 1ddbeaf first commit
```

_ Créez un alias de la commande précédente le nom de l'alias est : historique

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git config --global alias.historique "log --oneline --graph --decorate --all"
```

_ Affichez la liste des alias

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git config --get-regexp alias
alias.historique log --oneline --graph --decorate --all
```

_ Affichez l'historique des commit du fichier README.md avec l'alias.

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git historique README.md
* 1ddbeaf first commit
```

4 : Fourth Step : Excluding files

_ Renommer le fichier Licence.md à Licence.txt

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git mv Licence.md licence.txt
```

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

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git add licence.txt
```

_ Faites-le commit.

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "renommer"
[master bacc445] renommer
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename Licence.md => licence.txt (100%)
```

_ Créez un fichier nommé application.log (`touch application.log `)

_ Ne faites pas le stagging mais créez un fichier nommé « **.gitignore** » (`touch .gitignore `)

_ Sur le fichier **.gitignore** Ajoutez la ligne suivante « *.log » (code .)

_ Faites le stagging et le commit

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "log gitinore"
[master 028ba9f] log gitinore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git log
commit 028ba9f9f9263f22705f5b43b9b71cb773eb50a0 (HEAD -> master)
Author: othmane.qr <othmane.qorqach@gmail.com>
Date:   Tue Nov 26 15:12:19 2019 +0100

    log gitinore
```

_ **.gitignore** ignorer les fichiers *.log

5 : fifth Step : Branching and Merging

_ Modifiez le fichier README.md (code .)

_ Créez une branche pour la modification du fichier README.md du nom 'updates'

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git checkout -b updates
Switched to a new branch 'updates'

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (updates)
$
```

_ Faites le stagging et le commit en une seule ligne

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (updates)
$ git commit -am " add branch updates"
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
[updates 18c0488]  add branch updates
 1 file changed, 3 insertions(+), 1 deletion(-)

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (updates)
$
```

_ Affichez l'historique avec l'alias (git historique) .

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (updates)
$ git historique
* 18c0488 (HEAD -> updates)  add branch updates
* 028ba9f (master) log gitinore
* bacc445 renommer
* f171f6e add licence
* 1ddbeaf first commit
```

_ les modifications ont été apportées au branche Updates et HEAD devenue en branche Updates.

_ Retournez vers la branche Master (git checkout Master)

_ Affichez l'historique avec l'alias

```
$ git historique
* 18c0488 (updates)  add branch updates
* 028ba9f (HEAD -> master) log gitinore
* bacc445 renommer
* f171f6e add licence
* 1ddbeaf first commit
```

_ une fois retourné à un branche Master et tapé git historique nous trouvons HAED est devenue un branche Master, pas en branche Updates.

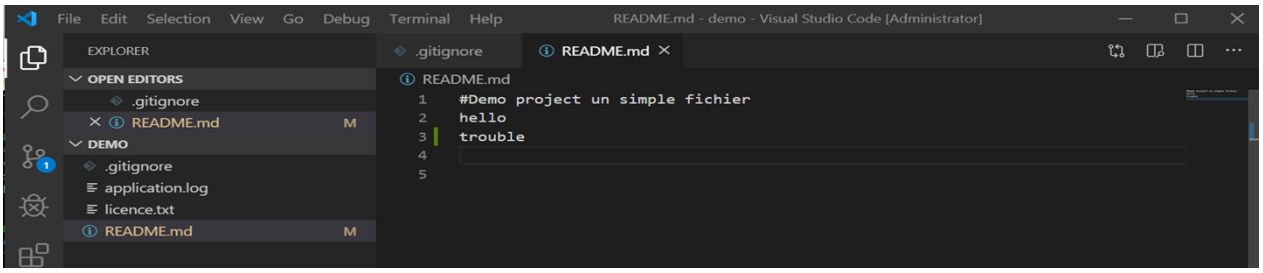
_ faites le merge

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git merge updates
Updating 028ba9f..18c0488
Fast-forward
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

6: sixth Step : Conflict Resolution

_ Créez une branche avec le nom ‘BAD’ (git checkout –b)

_ Modifiez le fichier README.md et ajouter la ligne ‘Trouble’ (code .)



_ Faites le stagging et le commit en une seule ligne (git commit –am ‘’….’’)

_retournez la branche Master (git checkout master)

_modifier le fichier README.md et ajouter la ligne ’Troubleshooting’ avec commande (code .)

_ Stagging/commiting avec commentaire ‘branche bien faite’ (git commit –am branche bien faite ’)

_ corriger le commentaire du dernier commit

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git commit --amend
[master e846b92] branche était bien faite
 Date: Wed Nov 27 13:42:57 2019 +0100
 1 file changed, 1 insertion(+), 1 deletion(-)
```

_ affichez historique

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git historique
* e846b92 (HEAD -> master) branche était bien faite
| * 545c2fa (BAD) add BAD
|/
* 18c0488 (updates)  add branch updates
* 028ba9f log gitinore
* bacc445 renommer
* f171f6e add licence
* 1ddbeaf first commit
```

_ nous avons trouvé un conflit, parce que n’est pas fusionner (merge)

_ Faites le merge de la branche ‘BAD’ (git merge BAD)

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git merge BAD
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

_ conflit n’est pas résoudre car il existe deux modification au même fichier, il doit être discuter avec nos équipe d’accord à seul modification.

_ la commande suivante : cat README.md

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ cat README.md
#Demo project un simple fichier
hello
<<<<<<< HEAD
Troubleshooting
=======
trouble

>>>>>>> BAD
```

La commande cat affichée le contenu de fichier.

_ la commande « git mergetool »

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge
araxis bc codecompare smerge emerge vimdiff
Merging:
README.md

Normal merge conflict for 'README.md':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff): |
```

Il faut installer un logiciel p4merge pour le GIT accepter la commande git mergetool.

7: seventh Step: merge tools

_ Installation de p4merge sur notre pc

_ tapez la commande git config – global --list

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config --global --list
user.name=othmane.qr
user.email=othmane.qorqach@gmail.com
alias.historique=log --oneline --graph --decorate --all

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$
```

_ Git config –global merge.tool p4merge

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config merge.tool p4merge
```

_ Git config –global mergetool.p4merge.path “lien d’installation”

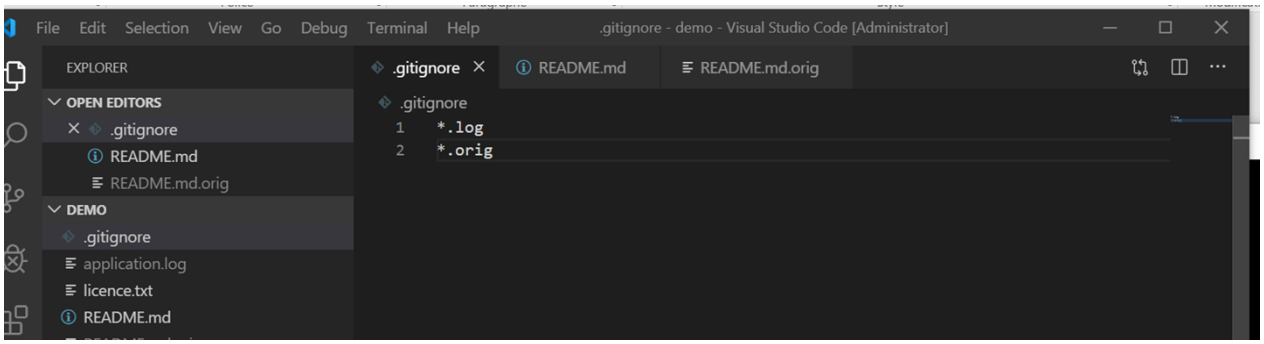
```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config --global mergetool.p4merge.path "C:\ProgramData\Microsoft\Windows\Start Menu\Pr
ograms\Perforce" .exe
```

_ configurer le prompt

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git config --global mergetool.prompt false

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git historique
*   347fe81 (HEAD -> master) merging
|\
| * 545c2fa (BAD) add BAD
* | e846b92 branche était bien faite
|/
* 18c0488 (updates)  add branch updates
* 028ba9f log gitinore
* bacc445 renommer
* f171f6e add licence
* 1ddbeaf first commit
```

8: eigth Step: Challenge

_ Sur le fichier **.gitignore** ; écrivez une clause pour rejeter les fichiers indésirables et (‘



On laisse que les fichiers : **licence.txt Readme.md .gitignore**

Scénario #2

1 : First Step : Tagging

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

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo (master)
$ git tag -a V1.0
```

```
REALEASE 1.0
#
# Write a message for tag:
#   V1.0
# Lines starting with '#' will be ignored.
~
~
~
~
~
```

_ Affichez les informations sur le TAG

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git show V1.0
tag V1.0
Tagger: othmane.qr <othmane.qorqach@gmail.com>
Date:   Thu Nov 28 09:44:47 2019 +0100

REALEASE 1.0

commit e40f9056b5ef35609c67e26f13dac9b45981e934 (HEAD -> master, tag: V1.0)
Author: othmane.qr <othmane.qorqach@gmail.com>
Date:   Wed Nov 27 16:28:16 2019 +0100

    finale

diff --git a/.gitignore b/.gitignore
index b5bd170..8f06263 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1 +1,2 @@
 *.log
+*.orig
\ No newline at end of file

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$
```

2 : Second Step : Stashing and Saving work in Progress

_ Modifiez le fichier README.md , ajoutez une ligne (code .)

_ la commande git Stash :

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git stash
Saved working directory and index state WIP on master: e40f905 finale

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$
```

Git Stash:

Nous utilisons git stash pour réserver notre changement dans une autre directory

_ Tapez la commande git stash list

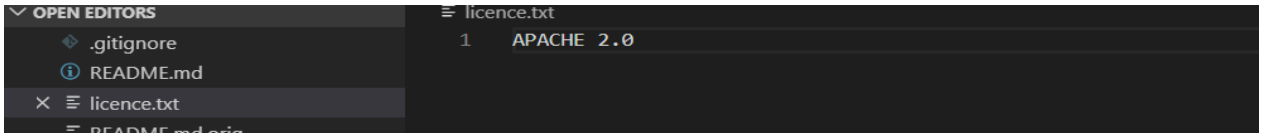
```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git stash list
stash@{0}: WIP on master: e40f905 finale

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$
```

_ Exécutez la commande git status

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git status
On branch master
nothing to commit, working tree clean
```

_ Nous constatons que la modification qu'on a fait est pas dans notre «working directory »

_ Modifiez le fichier « Licence.txt », ajoutez la ligne « APACHE 2.0 » (code .)



_ Faites stagging et le commit en une seule ligne

```
Youcode@DESKTOP-UAFPIDM MINGW64 ~/Desktop/projet/demo (master)
$ git commit -am "licence;txt mod"
[master 44251c1] licence;txt mod
 1 file changed, 1 insertion(+)
```

_ Exécutez la commande « git stash pop »

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
$ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (8c7e87c794814e6fe829e763bdddafab9c00a8ca)

Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo (master)
```

_ Afficher les changements faits dans l'autre directory (git stash)

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

_ Créez un repo github public sans ajouter le fichier README.md (github.com)

_ Examinez le remote

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo (master)
$ git remote add origin  https://github.com/yahyabouhlal/demo.git
```

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

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo (master)
$ git push -u origin master --tags
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (31/31), 2.93 KiB | 187.00 KiB/s, done.
Total 31 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/yahyabouhlal/demo.git
 * [new branch]      master -> master
 * [new tag]         V1.0 -> V1.0
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

4 : Fourth Step : Mini challenge (optionnel)

_ Créez un fichier sous git nommé **.SSH** **(mkdir .SSH)**

_Déplacez-vous dans le fichier .SSH (cd . SSH)

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo (master)
$ cd .SSH
```

Créez une authentification sécurisé SSH entre votre repo local et votre repo distant

```
Administrateur@DESKTOP-3K62L35 MINGW64 ~/Desktop/projects/demo/.SSH (master)
$ SSH-keygen -t rsa -C"othmane.qorqach@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrateur/.ssh/id_rsa):
Created directory '/c/Users/Administrateur/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrateur/.ssh/id_rsa.
Your public key has been saved in /c/Users/Administrateur/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:25byKCCv68NSDZxBAYcM7cF3zOvqndliZlUzG+s6UaI othmane.qorqach@gmail.com
The key's randomart image is:
+---[RSA 3072]----+
|=*+.             |
|.o=. . .         |
| o = . o         |
|  =      + . *   |
|   o    oSo + *  |
|  o o . Eoo.o    |
| o o o .o=+o     |
|. o . o O=o .    |
| o++    =..oo    |
+----[SHA256]-----+
```

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo/.SSH (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDYm3Io9mngI/hNZj5p4dog+fhyuXDwVMoGcdzzovZx
hLuYFK/u53zt8Zv7bYW/yCpdXa1huQS6x1f/Kcumdh3qCV+78uKHzFKLssjMAaZN4NpHPrKkviqADnoA
p44/SPLhBfpdXxAu+TuziSROed5OGaG1Ox0vSO3w3HN383b0mchFLCdBTHvNhGYzzjaLj7IDiBkrFOh6
3g3f6t7PL4BknqOSluNOfQCD42ww8TYORs5nmGTZNfg582YRSPka+RwPjCCSMF5cTcQpDXfJOjcbRmvB
VO2uewMRPct/W/hgJZTdtHhq1ORWM9nDouSESSHylCfNk8mLwPkzl6Hz2vryeYXbE4mC9+CjfeOcxwWW
Gl0/IwWERCRtSVlk/U6t5JQJy57Jq11F+5RZkBGibDBcc3mfBoL7+Gai6zpKAS+hTV5NkOmkE5atf1T1
WzGGvV3UpKCamSa78LEVFEnZvnzcQz4o3QfBjOOBJdbLdLT2N34rvT9pFrO7cGSXlB/WOOU= yahya.b
ouhlal.b@gmail.com
```

5: Fifth Step: Création d'une local copy

_ Sur github créez un autre repo nommé (Monsiteweb) (github.com)

_ Ajoutez à l'arboresence toujours sur github le fichier **.gitignore** et un fichier **licence.txt 'APACHE 2.0 '** sur (github.com)

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

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local (master)
$ git clone git@github.com:yahyabouhlal/monsiteweb.git
Cloning into 'monsiteweb'...
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 5.05 KiB | 1.26 MiB/s, done.
```

_ Vérifiez si le clone est créé.

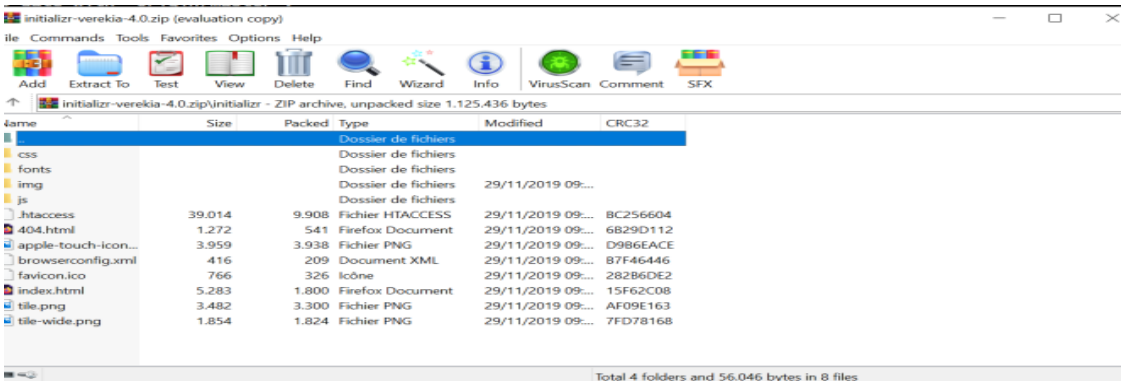
```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/demo/.SSH (master)
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
Hi yahyabouhlal! You've successfully authenticated, but GitHub does not provide shell
access.
```

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** (sur site)

_ Analysez l'arboresence.



l'arborescence :

c'est le contenu du site web (css ,fonts,img,js…)

_ Copiez le site télécharger dans votre repo local

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local
$ cp -R ../../initializr/* .
```

_ Git status

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        404.html
        apple-touch-icon.png
        browserconfig.xml
        css/
        favicon.ico
        fonts/
        index.html
        js/
        tile-wide.png
        tile.png

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

_ Faites le stagging et le commit en une seule ligne (git add –a && git commit -am)

_ Faite le push à github

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git push
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (30/30), 274.48 KiB | 1.53 MiB/s, done.
Total 30 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To github.com:yahyabouhlal/monsiteweb.git
   8967aff..7a12f71  master -> master
```

_ Vérifiez l'existence du site local sur votre repo github

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,et Faites le commit sur github (github.com)

_ Sur git et sur le repo local , Editez le fichier README.md (code .)

_ Faites le stagging et le commit en une seule ligne

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git commit -am "edit-md"
[master 44355e8] edit-md
 1 file changed, 2 insertions(+), 1 deletion(-)
```

les modification en add et commit dans une seul ligne parceque on deja dans locale repot

_ Executez la commande « git fetch »

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git fetch
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
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 github.com:yahyabouhlal/monsiteweb
   7a12f71..12b5589  master     -> origin/master
```

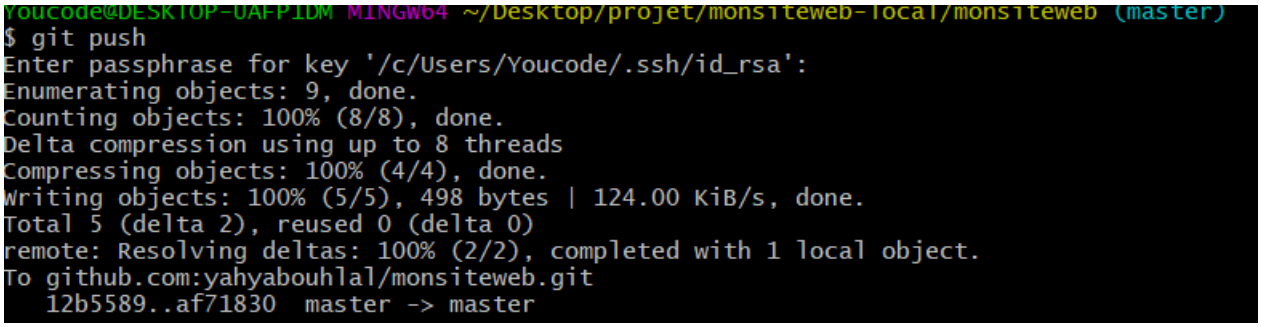
_ Git status

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (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
```

_ Git pull

```
Youcode@DESKTOP-UAFP1DM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git pull
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
Merge made by the 'recursive' strategy.
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

_ Git push


Youcode@DESKTOP-UAFPIDM MINGW64 ~/Desktop/projet/monsiteweb-local/monsiteweb (master)
$ git push
Enter passphrase for key '/c/Users/Youcode/.ssh/id_rsa':
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), 498 bytes | 124.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:yahyabouhlal/monsiteweb.git
12b5589..af71830 master -> master

(Scénario -3)

Etape 1 : Changes on Github

- L'ajout d'un fichier nommé style.css (touch style.css)
- Modification du fichier index.html en ajoutent labalise : (<h1>modification récente </h1>)
- Renommer le fichier 404.html en error404.html.
- La synchronisation avec le repo local
- Vérification de la liste du commit.
- Sur GITBASH, vérifiez avec « git status »
- Visualisez et télécharger les fichiers distants sur GITBASH.
- Faites le pull et fusionnez les changements distants avec le repo local. (git pull)(git merge).
- Affichez les informations du commit de l'ajout de la balise <h1> :

```
$ git log --follow -p -- index.html
commit 76a68704974cbbcbb0bf99de3e6b9ab7fbfb3574
Author: yahyabouhlal <57914696+yahyabouhlal@users.noreply.github.com>
Date:   Fri Nov 29 09:42:55 2019 +0100

    Update index.html

diff --git a/index.html b/index.html
index ce4736f..cd354a1 100644
--- a/index.html
+++ b/index.html
@@ -24,6 +24,8 @@
         <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]-->
```

Etape 2 : Branching and merging sur GITHUB

- La création du branch 'exemple' sur github.
- La modification du fichier README.md sur github Tjr.

- Dans le menu branch sur github on a un message du nouvelle branch ‘exemple’ pour faire un pull requeste.
- La création du branch ‘annulation’ sur repo local.
- Suppression du balise H1 dans index.html tjr sir repo local.
- Le commit et le stagging en une seule ligne.
- « Git show » pour vérifier vos changements.
- Pushez les changements qu’on a sur la branch annulation vers le repo distant.
- Nous constatons un message indiquant qu’une branche a été ajoutez avec un bouton « compare and pull request ».

Etape 3 : compare pull Requests

- La création d’un pull request avec le commentaire ‘annulation à vérifier’.
- Une page s’ouvre indique que la branch able to merge.
- Effectuer le merge.
- La suppression du branch annulation .

Etape 4 : merging en local

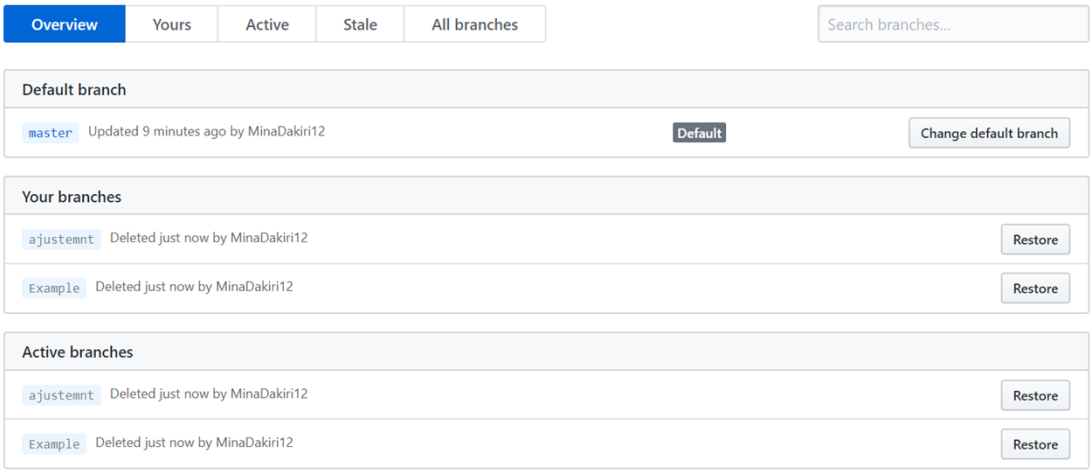
- La création du branch ajustement et lamodification du fichier main.css.
- Puchze les modifications sur Github.
- Revenez vers la branche master sur gitbash.
- Demande de pull (git pull).
- Type de merge on aura besoin pour faire un merge sans conflits : Fast-forward .

- Faites le merge à travers GITBASH :

```
youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb/css (ajustemnt)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb/css (master)
$ cd ..

youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git merge ajustemnt
Updating e318366..1141748
Fast-forward
 css/main.css | 9 ++++++++-
 index.html   | 2 +-
 2 files changed, 9 insertions(+), 2 deletions(-)
```

-
- Pushez les changements vers github.
- Explorez le pull pour visualiser les changements :

-
- Suppression des branchs qu'on a créé sur github :



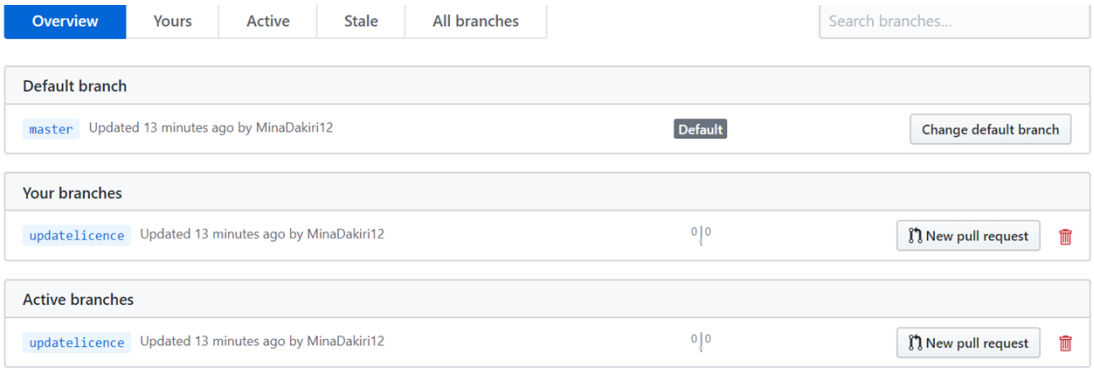
-
- Apporter les changements qu'on a fait sur GITHUB.
- Tapez la commande git branch –a : les branchs qu'on a supprimé sur github est resté dans le remote :

```
youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git branch -d annulation
Deleted branch annulation (was 16c5c25).

youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git branch -d ajustemnt
Deleted branch ajustemnt (was 1141748).

youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git branch -a
* master
  remotes/origin/Example
  remotes/origin/HEAD -> origin/master
  remotes/origin/ajustemnt
  remotes/origin/annulation
  remotes/origin/master
```

- La suppression des branchs qu'on a créé sur gitbash.
- Nous constatons que les branchs supprimer reste dans le remote.

Etape 5 : The Cleaning up

- 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 :

```
youcode@DESKTOP-UOM85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git push origin :updatelicence
To https://github.com/MinaDakiri12/Monsiteweb.git
 - [deleted]         updatelicence
```

- Supprimer la branch : Git branch –d 'updatelicence'.

- Exécutez la commande git branch –a : nous constatons que la branch qu'on a supprimé reste dans le remote.
- Trouvez la commande qui va nous permettre de supprimer la branche depuis le remote et ainsi de pushez les changements de la suppression simultanément : (git push origin –delete).

Etape 6 : Rebasing

- Editez le fichier README.md, ajoutez la ligne 'updates' et faite le commit.
- Sur GITBASH Editez le fichier index.html supprimez des lignes et modifier le titre du site.

```
<!--[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]><!-->
```

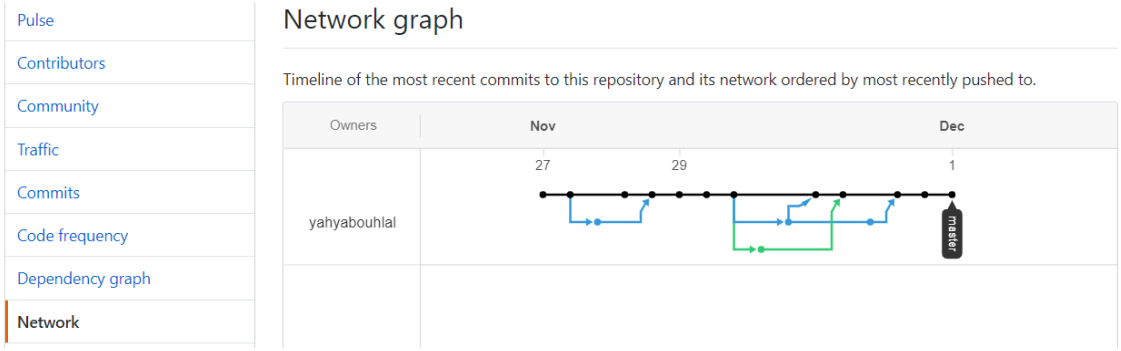
- Faites le stagging et le commit en une seule ligne.
- Maintenant apportez les changements faites sur le Repo distant : fetch.
- On a besoin de faire un rebase : git pull - - rebase.
- □ Exécutez l'alias que vous avez créé dans le scénario #1 (historique) :

```
youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git config --global alias.historique "log --oneline --decorate --graph --all"

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git historique
* 68cee9f (HEAD -> master) modification title and  ligne
* 3167317 (origin/master, origin/HEAD) ajoute line
* e1767a8 changed apache
* 2aa22b1 Update LICENSE
*   21a9ff3 Merge branch 'master' of https://github.com/MinaDakiri12/Monsiteweb
|\
| *   1a11115 Merge pull request #1 from MinaDakiri12/annulation
| |\
* | | 1141748 (origin/ajustemnt) modifier main.css
| |/
|/|
* | 16c5c25 (origin/annulation) supprimez la balise
* | e318366 modification in index.html
|/
| * ca67ecc (origin/Example) modification un message
|/
* 1a27d06 Rename 404.html to error 404 .html.
* 64e43f3 Update index.html
* a23d31a Create style.css
* 55815c6 modification
* c7f1710 adding website files
* 070f509 Initial commit
```

- Rebase permet de déplacer une branche et de changer son commit de départ (sa base).

Etape 7 : GitHub Insights

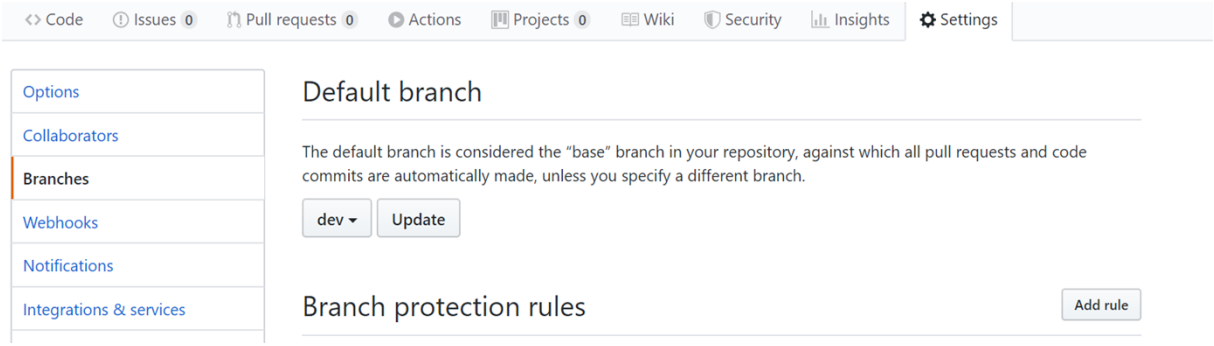
- Analysez le Network ou le timeline de votre workflow :



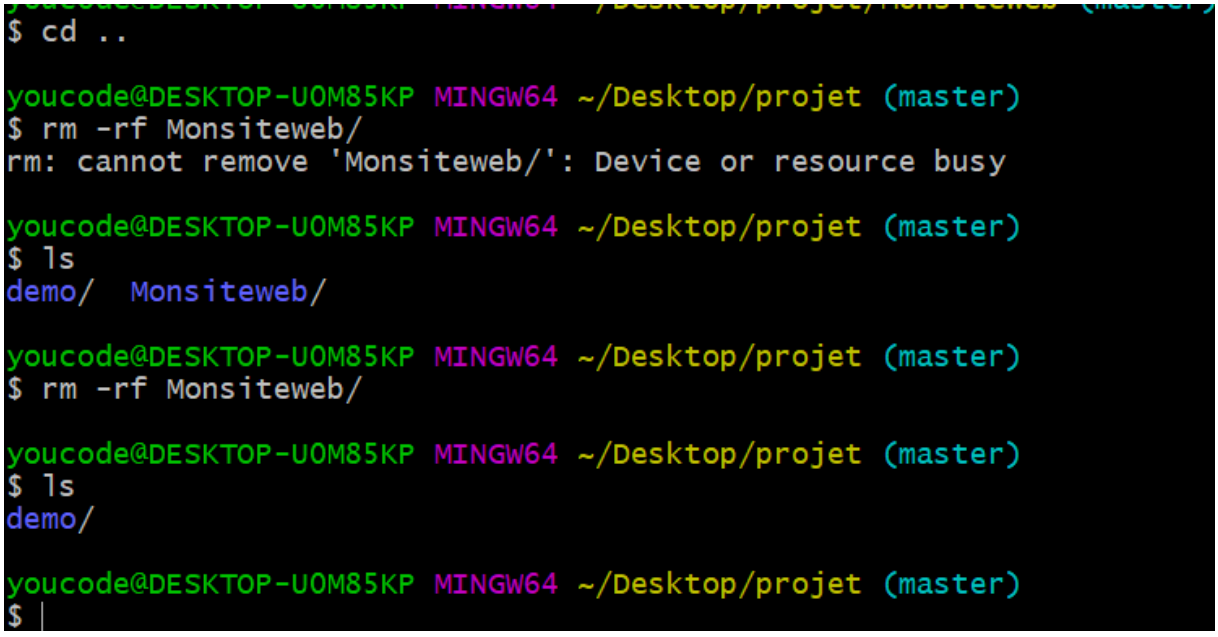
Etape 8 : Default branch and conflicts

- Création d'une branch nommé 'DIV' sur github.

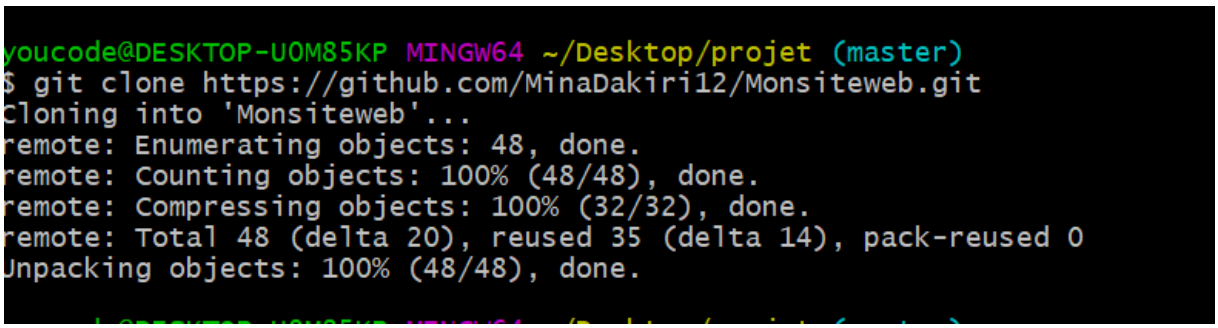
- La Branche Dev deviendra la branche par défaut ; sur le menu « settings » mettez la branche Dev en mode par défaut.



- Créez une autre branche « demo », on va la considérer comme « feature ».
- Supprimez la branche feature depuis GITHUB.
- Sur GIT, supprimez récursivement et localement votre repo website :



- Maintenant Clonez via https ou SSH



- Le contenu est récupéré :

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

-

```
youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (dev)
$ git checkout master
Switched to a new branch 'master'
Branch 'master' set up to track remote branch 'master' from 'origin'

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git switch
fatal: missing branch or commit argument

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (master)
$ git switch dev
Switched to branch 'dev'
Your branch is up to date with 'origin/dev'.

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (dev)
$ 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/MinaDakiri12/Monsiteweb
   3167317..66b6156  dev        -> origin/dev

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (dev)
$ vim README.md

youcode@DESKTOP-U0M85KP MINGW64 ~/Desktop/projet/Monsiteweb (dev)
$ git commit -am "modifier README.md"
[dev bfc3a86] modifier README.md
 1 file changed, 1 insertion(+), 1 deletion(-)
```