

12/04/2019

# Rapport

Brief\_Projet\_Scenario 1\_2\_3



**git**



Par Lamyaa Najahi

## **Sommaire :**

### INTRODUCTION

## *Scénario #1*

1-Immersion

2-Découverte

3-Historique

4-excluding files

5-Branching and

6-conflit résolution

7-mergetools :

8- challenge :

## *Scénario #2*

1 Tagging

2 Stashing and Saving work in Progress

3 Voyage sur Github, Local Repo to github Repo

4 Mini challenge ( optionnel)

5 Création d'une local copy :

6 Sending the website :

7 Fetch and pull :

## *Scénario #3*

*tableau trello*

# Introduction

Dans le cadre de la validation des compétences de la période SAS ; le brief projet est un moyen utile pour valider les compétences dans notre niveau respectif (N1, N2, N3)

**Dans ce projet nous avons travaillé pour plusieurs objectifs comme :**

- La formation d'un groupe de 5 apprenants
- Planification des tâches selon une méthode de gestion de projet (SCRUM).
- La finalisation des Scénarios 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

# Scénario #1

## #1 : First Step : Immersion

⇒ Définissez git en quelques mots ?

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds.

⇒ Le fonctionnement du fichier caché « .git ».

Le git repository (ou history tree) est le fichier .git habituellement caché, il contient l'historique des commits du projet. Il garde en mémoire tous les changements qui ont été comités. C'est ce qui est cloné quand on clone la version d'un "serveur".

⇒ Créez un répertoire nommé « /projects ».

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop (master)
⇒ $ mkdir test_git
```

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

```
⇒
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
⇒ $ mkdir demo
⇒
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
⇒ $ cd demo
```

⇒ Vérifiez le statut du répertoire « /demo ».

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
⇒ $ git status On branch master
⇒ No commits yet
⇒ nothing to commit (create/copy files and use "git add" to track)
```

⇒ Expliquez le commentaire.

⇒ Ce message signifie que notre copie de travail est propre, en d'autres mots, aucun fichier suivi n'a été modifié.

⇒ Créez le fichier README.md et ajouter la ligne suivante : '#Demo project un simple fichier'.

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ touch README.md
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ code .
```

⇒ Faites le staging et le committing avec un commentaire.

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git init
```

```
Initialized empty Git repository in C:/Users/Youcode/Desktop/test_git/demo/.git/
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
Git add .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git commit -m "first step"
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
README.md
```

```
nothing added to commit but untracked files present
```

## #2 : Second Step : La découverte

⇒ Déplacez-vous dans les fichiers de configuration « .git ».

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/demo (master)

⇒ \$ cd .git

⇒ Tapez Ls -al

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/demo/.git (GIT\_DIR!)

⇒ \$ ls -al

⇒ total 13

⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 ./

⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 ../

⇒ -rw-r--r-- 1 Youcode 197121 11 déc. 2 10:08 COMMIT\_EDITMSG

⇒ -rw-r--r-- 1 Youcode 197121 130 déc. 2 10:08 config

⇒ -rw-r--r-- 1 Youcode 197121 73 déc. 2 10:08 description

⇒ -rw-r--r-- 1 Youcode 197121 23 déc. 2 10:08 HEAD

⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 hooks/

⇒ -rw-r--r-- 1 Youcode 197121 65 déc. 2 10:08 index

```

⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 info/
⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 objects/
⇒ drwxr-xr-x 1 Youcode 197121 0 déc. 2 10:08 refs/

```

⇒ Expliquez les clauses suivantes : HEAD, LOGS, BRANCHES

- HEAD : les informations sur la branche et les commit qu'il contienne etc.

- LOG : l'historique des commits

- BRANCH : un moyen de demander un nouveau répertoire de travail, une nouvelle zone de staging et un nouvel historique de projet.

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

```

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo/.git (GIT_DIR!)
⇒ $ cd ..
⇒
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
⇒ $ touch Licence.md

```

⇒ Faites-le Commit

```

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
⇒ $ git add .

```

```

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
⇒ $ git commit -m "second stepp"
⇒ On branch master
⇒
⇒ Initial commit
⇒
⇒ Untracked files:
⇒   Licence.md
⇒   README.md
⇒
⇒ nothing added to commit but untracked files present

```

⇒ Affichez les fichiers traqués.

```

⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
⇒ $ git ls-tree -r master --name-only
⇒ 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

```

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git log --all --decorate --oneline --graph
* e549918 (HEAD -> master) second step

```

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

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git config --global alias.historique "log --all --decorate --oneline --graph"
```

⇒ Affichez la liste des alias

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git config --global --list | grep alias
```

```
alias.historique=log --all --decorate --oneline --graph
```

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

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
⇒ $ git historique README.md
```

```
⇒ * e549918 (HEAD -> master) second step
```

## #4 : Fourth Step : Excluding files

⇒ Renommer le fichier Licence.md à Licence.txt

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
⇒ $ mv Licence.md Licence.txt
```

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

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git add -u
```

⇒ Faites-le commit.

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
⇒ $ git commit -m "fourth step"
```

```
⇒ [master 92c3740] fourth step
```

```
⇒ 1 file changed, 0 insertions(+), 0 deletions(-)
```

```
⇒ delete mode 100644 Licence.md
```

```
⇒
```

après ça ne faites pas le staging et nous avons créé un fichier nommé « .gitignore » et sur le fichier .gitignore nous avons Ajouter la ligne \*.log » et Faites le staging et le commit

Voilà le résultat de ce que nous avons fait :

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ touch application.log
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ touch .gitignore
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ echo "*.log" > .gitignore
*.log .gitignore
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ cat
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ echo "*.log" > .gitignore
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ cat
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ cat .gitignore
*.log
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git add .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git commit -m "step 4"
[master b492d88] step 4
 2 files changed, 1 insertion(+)
 create mode 100644 .gitignore
 create mode 100644 Licence.txt
```

## #5 : fifth Step : Branching and Merging

Dans cette etape nous avons fait des modifications sur le fichier readme.md au niveau de branche master et au niveau d'une nouvelle branche nommé 'updates'.

Voila la résultat :

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git checkout -b updates
```



Switched to a new branch 'updates'

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (updates)
```

```
$ git commit -am "step fifth"
```

```
[updates 80ede8f] step fifth
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (updates)
```

```
$ git historique
```

```
* 80ede8f (HEAD -> updates) step fifth
```

```
* b492d88 (master) step 4
```

```
* 92c3740 fourth step
```

```
* e549918 second step
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git merge updates
```

```
Updating b492d88..80ede8f
```

```
Fast-forward
```

```
README.md | 3 ++-
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

Et en fin nous constaté que la modification qu'on fait avant la création de la branche updates, se trouve dans cette dernière.

## #6: sixth Step : Conflict Resolution

Dans cette étape nous avons créé une branche avec le nom 'BAD' et modifié le fichier README.md pour ajouter la ligne 'Trouble' et quelques autres étapes et le résultat est comme ça

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git checkout -b BAD
```

```
Switched to a new branch 'BAD'
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (BAD)
```

```
$ code .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (BAD)
```

```
$ git commit -am "sixth step"
```

```
[BAD 28fa938] sixth step
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (BAD)
```

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ code .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git commit -am "branche bien faite"
[master ce5c725] branche bien faite
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git commit --amend
[master 70ecf76] branche bien faite corrigé
Date: Mon Dec 2 11:46:48 2019 +0100
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git log
commit 70ecf76a777ec791d1bb3aefa6cd70b4eeb66a8d (HEAD -> master)
Author: Lamyaa <lamyaaajahi96@gmail.com>
Date: Mon Dec 2 11:46:48 2019 +0100
```

branche bien faite  
corrigé

```
commit 80ede8fe1752f0479a6b4595de55fb6c16a3f3d4 (updates)
Author: Lamyaa <lamyaaajahi96@gmail.com>
Date: Mon Dec 2 11:37:07 2019 +0100
```

step fifth

```
commit b492d880083e56ea92fc78528a8eaf49a28568f6
Author: Lamyaa <lamyaaajahi96@gmail.com>
Date: Mon Dec 2 11:12:34 2019 +0100
```

step 4

```
commit 92c3740896b0d615dbfecb84058ee21fa3b04578
Author: Lamyaa <lamyaaajahi96@gmail.com>
Date: Mon Dec 2 11:02:51 2019 +0100
```

fourth step

```
commit e5499187e4dec66ab1ff2e134b592e82471975d2
Author: Lamyaa <lamyaaajahi96@gmail.com>
Date: Mon Dec 2 10:27:43 2019 +0100
```

second step

(END)

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
bash: oiiiiinnfd,gnf: command not found
bash: dng: command not found
bash: fngk: command not found
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/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.
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master|MERGING)
```

```
$ cat README.md
#Demo project un simple fichier'.a
step fifth test
<<<<<<< HEAD
Troubleshooting
=====
Trouble
>>>>>>> BAD
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master|MERGING)
```

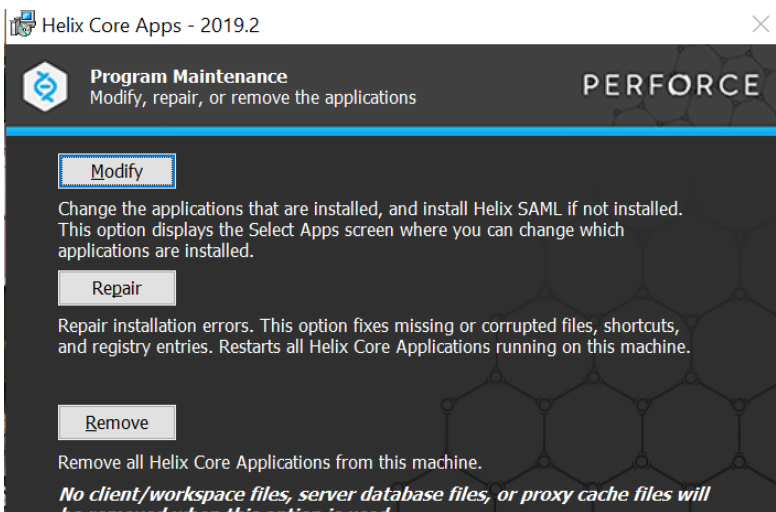
```
$ git mergetool
Merging:
README.md
```

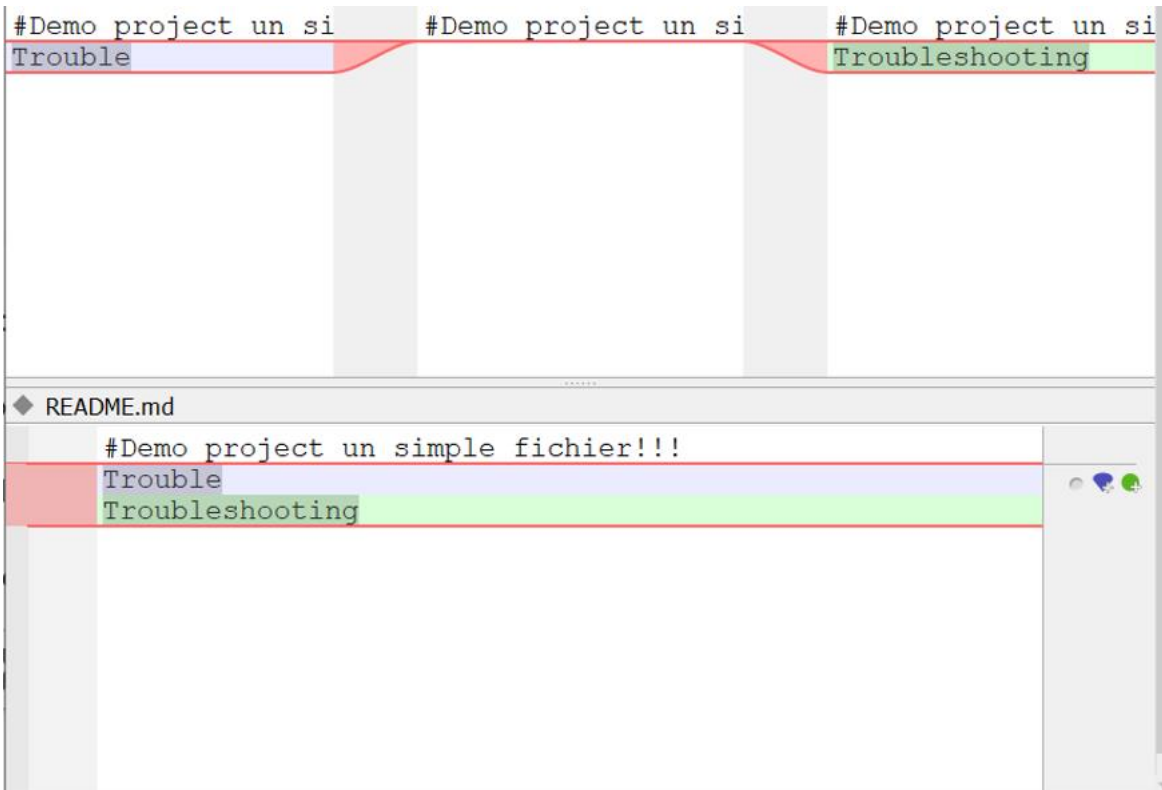
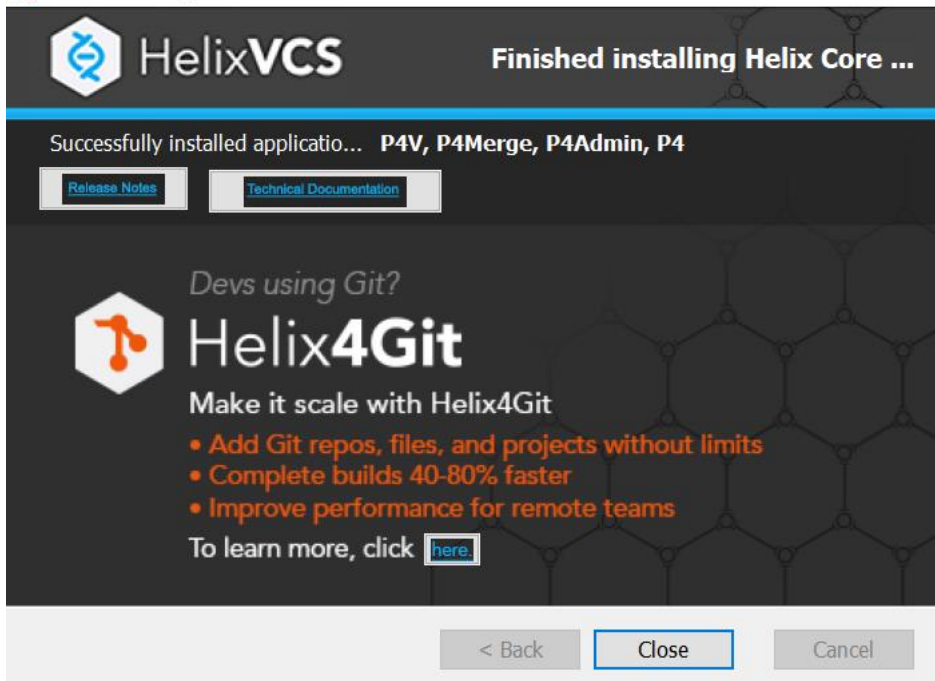
```
Normal merge conflict for 'README.md':
  {local}: modified file
  {remote}: modified file
```

Après le fusionnement on voit un message de conflit à cause de modification du même fichier README .md dans les deux branches

## #7: seventh Step: merge tools

Ici nous installer le programme p4merge et après l'installation, On a configuré git on ajoutant p4merge comme un mergetool...

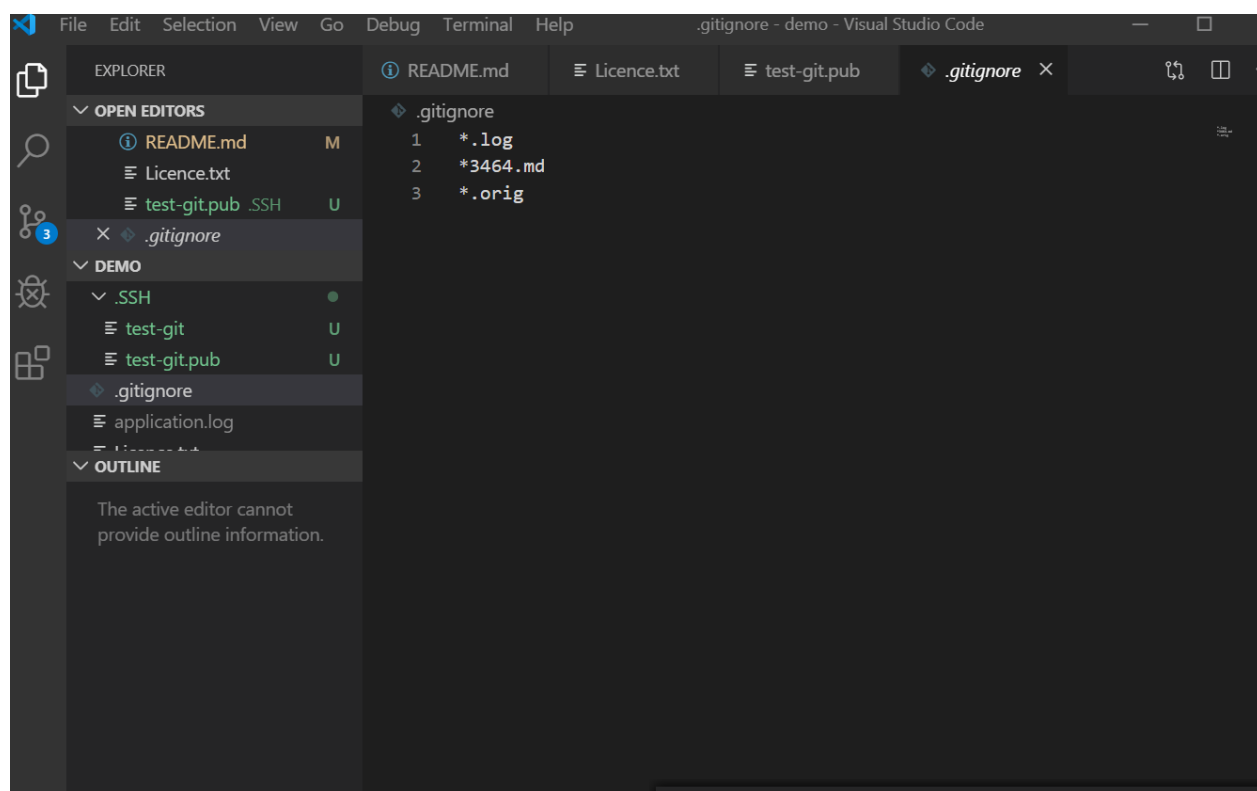




Après l'installation et après écrit la commande « git mergetool » logiciel affiche un interface pour montre les conflits.

## #8: eighth Step: Challenge

- ⇒ Finalement dans cette étape Sur le fichier `.gitignore` ; nous avons écrite une clause pour rejeter les fichiers indésirables et redondants



# Scénario #2

## #1 : First Step : Tagging

nous Avons vue comment créer un tag et comment afficher son information et leur utilisation c'est le tag comme un référence un commit particulie.

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git tag V1.0 -m "REALEASE 1.0"
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git show
commit 263eba4db7246ea170a79712011da5d17b6fb6e7 (HEAD -> master, tag: V1.0)
Merge: 70ecf76 28fa938
Author: Lamyaa <lamyaaanajahi96@gmail.com>
Date: Mon Dec 2 13:43:24 2019 +0100
```

fin commit

```
diff --cc .gitignore
index 397b4a7,397b4a7..327adf7
--- a/.gitignore
+++ b/.gitignore
@@@ -1,1 -1,1 +1,3 @@@
*.log
++*3464.md
++*.orig
diff --cc README.md
index ba389c1,1c16dc8..2590e96
--- a/README.md
+++ b/README.md
@@@ -1,3 -1,3 +1,2 @@@
#Demo project un simple fichier'.a
--step fifth test
-Trouble
+Troubleshooting
```

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

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ code .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git stash
```

```
Saved working directory and index state WIP on master: 263eba4 fin commit
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git stash list
```

```
stash@{0}: WIP on master: 263eba4 fin commit
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ code .
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ git commit -am "seconde scenario"
```

```
[master edb1298] seconde scenario
```

```
1 file changed, 1 insertion(+)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/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} (b00f7ac2794091c60b529022f128a91d747fad97)
```


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

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


---

Owner

Repository name \*

 lamyita


 / 

test\_git 


Great repository names are short and memorable. Need inspiration? How about [solid-octo-computing-machine](#)?

Description (optional)

---

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.


---

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.


Add .gitignore: **None**

Add a license: **None** 

**Create repository**

⇒ Créez le remote en https

**Quick setup — if you've done this kind of thing before**


 Set up in Desktop

 or 

HTTPS

SSH


https://github.com/lamyita/test\_git2.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


**...or create a new repository on the command line**

```
echo "# test_git2" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/lamyita/test_git2.git
git push -u origin master
```



**...or push an existing repository from the command line**

```
git remote add origin https://github.com/lamyita/test_git2.git
git push -u origin master
```



⇒ Examinez le remote

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git remote add origin https://github.com/lamyita/test-git.git
```

⇒ Poussez le tout à travers la commande : `git push -u origin master -tags`

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
$ git push -u origin master --tags
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (26/26), 2.18 KiB | 186.00 KiB/s, done.
Total 26 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/lamyita/test-git.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)

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ mkdir .SSH
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo (master)
```

```
$ cd .SSH
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/demo/.SSH (master)
```

```
$ ssh-keygen
```

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Youcode/.ssh/id\_rsa): test-git

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in test-git.

Your public key has been saved in test-git.pub.

The key fingerprint is:

SHA256:zGLOJE5GZd7v9cK9uoyQqVBRZs35dTI7tTNbpQy2gg0 Youcode@DESKTOP-M6PAE21

The key's randomart image is:

```
+---[RSA 3072]-----+
|      o +o .      |
|      + =E + o + +|
|      . o .+ o = B o|
|      .   +..o o =+.|
|      + = S ... . =|
|      + B . + o o . |
|      o o + . o o  |
|      . . . o . .  |
|      .   . +o.   |
+-----[SHA256]-----+
```

## #5: Fifth Step: Création d'une local copy :

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner

Repository name \*

 lamyita /  

Great repository names are short and memorable. Need inspiration? How about **animated-eureka**?

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

- ⇒ Après cette étape nous avons ajouté à l'arborescence toujours sur github le fichier **.gitignore** et un fichier **licence.txt** 'APACHE 2.0' et nous avons créé un clone github vers le local sous le nom (Monsiteweb-local)

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
$ git clone https://github.com/lamyita/-Monsiteweb.git
Cloning into '-Monsiteweb'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), done.
```

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

- ⇒ Télécharger le site web depuis le lien suivant : <http://www.initializr.com/>  
⇒ Sur le site télécharger un site bootstrap avec le fichier **.htaccess** et le fichier **404.html**

---

### 1 - Pre-configuration

**Classic H5BP**  
Docs Demo

**Responsive**  
Docs Demo

**Bootstrap**  
Docs Demo

2) Sur le site télécharger un site bootstrap avec le fichier **.htaccess** et le fichier **404.html**

| HTML/CSS Template                             | HTML5 Polyfills   | jQuery                                       |
|---|---|--|
| <input checked="" type="radio"/> No template  | <input type="radio"/> Modernizr                                   | <input checked="" type="checkbox"/> Minified |
| <input type="radio"/> Mobile-first Responsive | <input checked="" type="radio"/> Just HTML5shiv                   | <input type="checkbox"/> Development         |
| <input type="radio"/> Twitter Bootstrap       | <input checked="" type="checkbox"/> Respond - <b>Alternatives</b> |  |

### H5BP Optional

|   |  |  |
|---|--|--|
| <input type="checkbox"/> IE Classes           | <input type="checkbox"/> Favicon                 | <input type="checkbox"/> Humans.txt          |
| <input type="checkbox"/> Old browser warning  | <input type="checkbox"/> Apple and Windows Icons | <input checked="" type="checkbox"/> 404 Page |
| <input type="checkbox"/> Google Analytics     | <input type="checkbox"/> plugins.js              | <input type="checkbox"/> Adobe Cross Domain  |
| <input checked="" type="checkbox"/> .htaccess | <input type="checkbox"/> Robots.txt              |  |

**Download it!**

**What's inside?**

PC > Bureau > initializr

Rechercher dans : initializr

| <input type="checkbox"/> Nom | Modifié le       | Type                 | Taille |
|------------------------------|------------------|----------------------|--------|
| css                          | 27/11/2019 16:06 | Dossier de fichiers  |        |
| img                          | 27/11/2019 13:06 | Dossier de fichiers  |        |
| js                           | 27/11/2019 16:06 | Dossier de fichiers  |        |
| .editorconfig                | 27/11/2019 13:06 | Fichier EDITORCONFIG | 1 Ko   |
| .gitattributes               | 27/11/2019 13:06 | Document texte       | 1 Ko   |
| .gitignore                   | 27/11/2019 13:06 | Document texte       | 1 Ko   |
| .htaccess                    | 27/11/2019 13:06 | Fichier HTACCESS     | 39 Ko  |
| 404                          | 27/11/2019 13:06 | Firefox Document     | 2 Ko   |
| index                        | 27/11/2019 13:06 | Firefox Document     | 1 Ko   |

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

```
⇒ Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
⇒ $ cp -R ~/Desktop/initializr/* .
⇒
```

Et les autres question dans cette étape

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
```

```
$ ls
```

```
404.html          css/              fonts/           js/              tile-wide.png
apple-touch-icon.png demo/             img/             -Monsiteweb/
browserconfig.xml favicon.ico       index.html      tile.png
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
```

```
$ ls
```

```
demo/ -Monsiteweb/
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git (master)
```

```
$ git status
```

```
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'Menu Démarrer/': Permission denied
warning: could not open directory 'Mes documents/': Permission denied
warning: could not open directory 'Modèles/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Voisinage d'impression/': Permission denied
warning: could not open directory 'Voisinage réseau/': Permission denied
```

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

```
deleted:    ../Lamyaa/demo/README.md
deleted:    ../desktop.ini
deleted:    ../projet
deleted:    ../test_branche.PNG
```

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git (master)

\$ cd Monsiteweb/

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/Monsiteweb (master)

\$ git commit -am "sixth\_step\_scenario\_2"

On branch master

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

Untracked files:

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/Monsiteweb (master)

\$ git add . && git commit -m "sixth\_step\_scenario\_2"

warning: LF will be replaced by CRLF in 404.html.

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

warning: LF will be replaced by CRLF in browserconfig.xml.

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

warning: LF will be replaced by CRLF in css/bootstrap-theme.css.

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

warning: LF will be replaced by CRLF in css/bootstrap-theme.min.css.

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

warning: LF will be replaced by CRLF in css/bootstrap.css.

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

warning: LF will be replaced by CRLF in css/bootstrap.min.css.

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

warning: LF will be replaced by CRLF in css/main.css.

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

warning: LF will be replaced by CRLF in fonts/glyphicons-halflings-regular.svg.

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

warning: LF will be replaced by CRLF in index.html.

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

warning: LF will be replaced by CRLF in js/main.js.

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

warning: LF will be replaced by CRLF in js/vendor/bootstrap.js.

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

warning: LF will be replaced by CRLF in js/vendor/bootstrap.min.js.

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

warning: LF will be replaced by CRLF in js/vendor/jquery-1.11.2.min.js.

The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in js/vendor/modernizr-2.8.3-respond-1.4.2.min.js.

The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in js/vendor/npm.js.

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

[master c31f64b] sixth\_step\_scenario\_2

24 files changed, 9599 insertions(+)

create mode 100644 404.html

create mode 100644 apple-touch-icon.png

create mode 100644 browserconfig.xml

create mode 100644 css/bootstrap-theme.css

create mode 100644 css/bootstrap-theme.css.map

create mode 100644 css/bootstrap-theme.min.css

create mode 100644 css/bootstrap.css

create mode 100644 css/bootstrap.css.map

create mode 100644 css/bootstrap.min.css

create mode 100644 css/main.css

create mode 100644 favicon.ico

create mode 100644 fonts/glyphicons-halflings-regular.eot

create mode 100644 fonts/glyphicons-halflings-regular.svg

create mode 100644 fonts/glyphicons-halflings-regular.ttf

create mode 100644 fonts/glyphicons-halflings-regular.woff

create mode 100644 index.html

create mode 100644 js/main.js

create mode 100644 js/vendor/bootstrap.js

create mode 100644 js/vendor/bootstrap.min.js

create mode 100644 js/vendor/jquery-1.11.2.min.js

create mode 100644 js/vendor/modernizr-2.8.3-respond-1.4.2.min.js

create mode 100644 js/vendor/npm.js

create mode 100644 tile-wide.png

create mode 100644 tile.png

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/Monsiteweb (master)

\$ git push

Enumerating objects: 30, done.

Counting objects: 100% (30/30), done.

Delta compression using up to 8 threads

Compressing objects: 100% (29/29), done.

Writing objects: 100% (29/29), 274.48 KiB | 2.23 MiB/s, done.

Total 29 (delta 3), reused 0 (delta 0)

remote: Resolving deltas: 100% (3/3), done.

To https://github.com/lamyita/-Monsiteweb.git

0a10ddf..c31f64b master -> master

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

Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test\_git/Monsiteweb (master)

\$ code .

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/Monsiteweb (master)
$ git commit -am "seventh step"
[master 7bd241d] seventh step
 1 file changed, 2 insertions(+), 1 deletion(-)
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/Monsiteweb (master)
$ git fetch
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/lamyita/-Monsiteweb
   c31f64b..83c0af7  master    -> origin/master
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/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

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/Monsiteweb (master)
$ git pull
Merge made by the 'recursive' strategy.
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Youcode@DESKTOP-M6PAE21 MINGW64 ~/Desktop/test_git/Monsiteweb (master)
$ git push
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), 573 bytes | 191.00 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/lamyita/-Monsiteweb.git
   83c0af7..9012be5  master -> master
```

## Scénario #3

Il y a 8 étapes

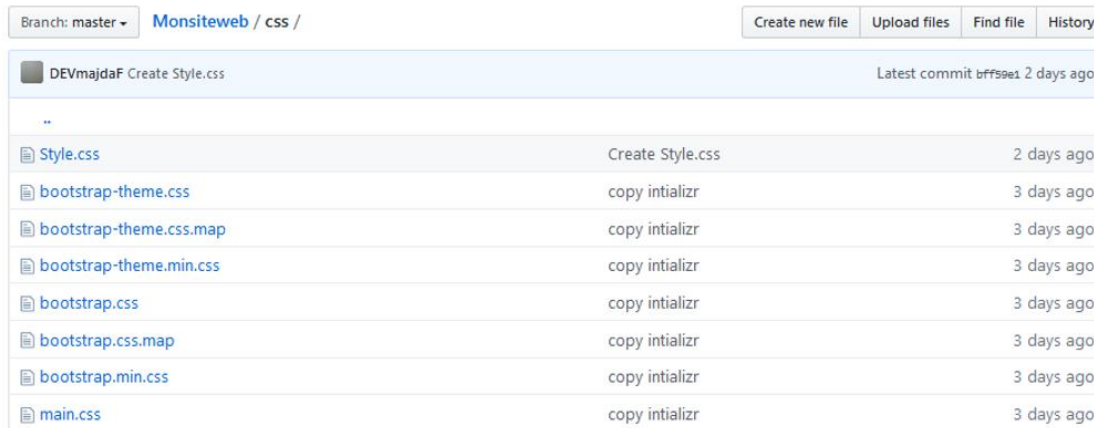
Parmi les choses qui nous avons fait la différence entre git fetch et git pull.

La commande git fetch va récupérer toutes les données des commits effectués sur la branche courante qui n'existent pas encore dans votre version en local. Ces données seront stockées dans le répertoire de travail local mais ne seront pas fusionnées avec votre branche locale.

La commande git pull est en fait la commande qui regroupe les commandes git fetch suivie de git merge. Cette commande télécharge les données des commits qui n'ont pas encore été récupérées dans votre branche locale puis fusionne ensuite ce

données. Voici quelques commandes dans le scénario 3

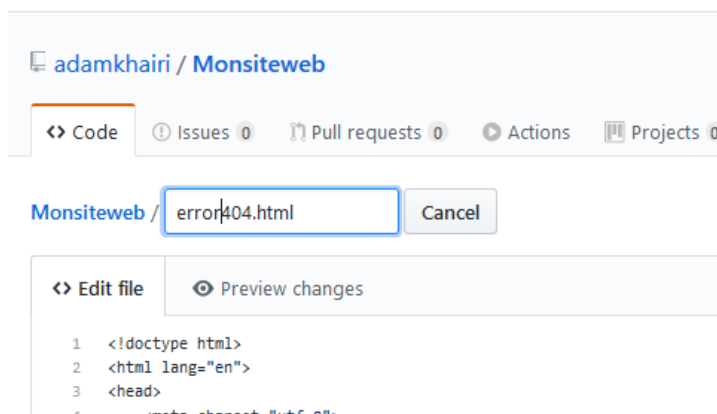
**Sur le dossier CSS ajoutez un autre fichier nommé style.css**



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

```
24     <script src="js/vendor/modernizr-2.8.3-respons-1
25 </head>
26 <body>
27   <h1> modification récente </h1>
28   <!--[if lt IE 8]>
29     <p class="browserupgrade">You are using an <
```

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





## Sur gitHub vérifiez la liste des commits .git

|   |          |         |
|---|----------|---------|
| Commits on Dec 1, 2019  |          |         |
| Rename 404.html to error404.html<br>adamkhairi committed 37 seconds ago   | Verified | 70c1d9f |
| Commits on Nov 30, 2019   |          |         |
| add H1<br>adamkhairi committed 14 minutes ago   | Verified | d29e347 |
| Merge pull request #1 from adamkhairi/add-license-1 ...<br>adamkhairi committed 21 minutes ago                        | Verified | 9a6fa78 |
| Commits on Nov 29, 2019   |          |         |
| Create Style.css<br>DEVmajdaF committed 2 days ago  | Verified | bff59e1 |
| Commits on Nov 27, 2019   |          |         |
| Message! Merge branch 'master' of https://github.com/DEVmajdaF/Monsiteweb-local ...<br>DEVmajdaF committed 3 days ago |          | 4a29a9d |
| editing Readme.md<br>DEVmajdaF committed 3 days ago   |          | 5795658 |
| Update title<br>DEVmajdaF committed 3 days ago  | Verified | 33535a8 |
| copy intializr<br>DEVmajdaF committed 3 days ago  |          | ff069e0 |
| Create LICENSE.txt<br>DEVmajdaF committed 3 days ago  | Verified | d654e3c |
| Create .gitignore<br>DEVmajdaF committed 3 days ago   | Verified | 9f63574 |
| Initial commit<br>DEVmajdaF committed 3 days ago  | Verified | 687558c |

## 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
```

## 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
```

**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%)
```

## *tableau trello*

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011, et inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

La version de base est gratuite, tandis qu'une formule payante permet d'obtenir des services supplémentaires. Le service est disponible en plusieurs langues (23 en juin 2016).

