



Scénarios

Rapport

Réalisé par :

- Abdelghafour rochdi
- Jihad karamane
- Elkassir hamza
- Souad eddibes
- Yassine moumen

Liens github :

<https://github.com/AbdelRochdi/brief>

<https://github.com/AbdelRochdi/Monsiteweb>

Scénario 1 :

Product BACKLOG :

1 : First Step : Immersion

⇒ Définissez git en quelques mots ?

Git est un outil de bas niveau, simple et performant, qui a pour but principal de gérer l'évolution d'une arborescence de fichiers.

⇒ Expliquez le fonctionnement du fichier caché « `.git` ».

`.git` est un dossier caché situé à la racine du projet. Il contient l'historique des modifications des fichiers et la configuration de Git pour ce projet.

⇒ Sur votre bureau créez un répertoire nommée « `/projects` » : `mkdir projects`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop  
$ mkdir projects
```

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

```
mkdir demo
```

```
git init
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop  
$ cd projects  
  
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects  
$ mkdir demo  
  
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects  
$ cd demo  
  
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo  
$ git init  
Initialized empty Git repository in c:/users/youcode/Desktop/projects/demo/
```

⇒ Vérifiez le statut du répertoire « `/demo` ».

```
git status
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/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.

No commits yet

nothing to commit (create/copy files and use "git add" to track)

rien a commiter pour le moment

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

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ nano README.md
```

- ⇒ Faites le staging et le commiting avec un commentaire.

git add . && commit -m « commit 1 »

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git add -A && git commit -m "commit 1"
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
[master (root-commit) 043b401] commit 1
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

2 : Second Step : La découverte

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

cd .git

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ cd .git
```

- ⇒ Tapez Ls -al

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo/.git (GIT_DIR!)
$ ls -al
total 13
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:27 .
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:26 ..
-rw-r--r-- 1 youcode 197121 9 déc. 2 11:27 COMMIT_EDITMSG
-rw-r--r-- 1 youcode 197121 130 déc. 2 11:21 config
-rw-r--r-- 1 youcode 197121 73 déc. 2 11:21 description
-rw-r--r-- 1 youcode 197121 23 déc. 2 11:21 HEAD
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:21 hooks/
-rw-r--r-- 1 youcode 197121 137 déc. 2 11:27 index
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:21 info/
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:27 logs/
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:27 objects/
drwxr-xr-x 1 youcode 197121 0 déc. 2 11:21 refs/
```

- ⇒ Expliquez les clauses suivantes : HEAD, LOGS, BRANCHES

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

```
cd ..
```

```
touch licence.md
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo/.git (GIT_DIR!)
$ cd ..

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ touch licence.md
```

- ⇒ Faites-le Commit

```
git add .
```

```
git commit -m «commit 2»
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git add .

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "commit 2"
[master 59160dc] commit 2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 licence.md
```

- ⇒ Affichez les fichiers traqués.

```
ls
```

```
youcode@DESKTOP-ORTAJ96 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

```
git log --oneline --graph --decorate --all
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git log --oneline --graph --decorate --all
* 59160dc (HEAD -> master) commit 2
* 043b401 commit 1
```

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

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

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git config --global alias.historique "log --oneline --graph --decorate --all"
```

- ⇒ Affichez la liste des alias

```
git config --get-regexp alias
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git config --get-regexp alias
alias.historique log --oneline --graph --decorate --all
alias.historique2 reflog
```

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

```
git historique README.md
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git historique README.md
* 043b401 commit 1
```

4 : Fourth Step : Excluding files

- ⇒ Renommer le fichier Licence.md à Licence.txt

```
git mv licence.md licence.txt
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git mv licence.md licence.txt
```

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

```
git add licence.txt
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git add licence.txt
```

- ⇒ Faites-le commit.

```
git commit -m "commit 3"
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "commit 3"
[master 287b06c] commit 3
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename licence.md => licence.txt (100%)
```

- ⇒ Supposons que nous développons sur notre plateforme, sûrement on a des fichiers qu'on veut exclure de notre arborescence du repository Local

1. Créez un fichier nommé application.log

```
touch application.log
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ touch application.log
```

2. Ne faites pas le staging mais créez un fichier nommé « **.gitignore** »

```
touch .gitignore
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ touch .gitignore
```

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

```
nano .gitignore
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ nano .gitignore
```

4. Faites le staging et le commit.

```
git add .
```

```
git commit -m «commit 4»
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/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-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -m "commit 4"
[master 67a08a8] commit 4
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

5. Qu'est-ce que vous constatez ?

on constate que le fichier application.log a été ignoré par le fichier **.gitignore**, il est donc impossible de l'ajouter sauf par utilisation de **-f**.

5 : fifth Step : Branching and Merging

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

- ⇒ The fast forward merging
- ⇒ The manual
- ⇒ The automatic
- ⇒ Modifiez le fichier README.md

```
nano README.md
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ nano README.md
```

- ⇒ Ne faites pas le commit
- ⇒ Créez une branche pour la modification du fichier README.md du nom 'updates'

```
git branch updates
```

```
git checkout updates
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git branch updates

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git checkout updates
Switched to branch 'updates'
M       README.md
```

⇒ Faites le staging et le commit en une seule ligne : `git commit -am "commit 5"`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (updates)
$ git commit -am "commit 5"
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
[updates dd0ea54] commit 5
 1 file changed, 1 insertion(+)
```

⇒ Affichez l'historique avec l'alias

`git historique`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (updates)
$ git historique
* dd0ea54 (HEAD -> updates) commit 5
* 67a08a8 (master) commit 4
* 287b06c commit 3
* 59160dc commit 2
* 043b401 commit 1
```

⇒ Qu'est-ce que vous constatez ?

⇒ Retournez vers la branche Master :

`git checkout master`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (updates)
$ git checkout master
Switched to branch 'master'
```

⇒ Affichez l'historique avec l'alias

`git historique`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git historique
* dd0ea54 (updates) commit 5
* 67a08a8 (HEAD -> master) commit 4
* 287b06c commit 3
* 59160dc commit 2
* 043b401 commit 1
```

⇒ Qu'est-ce que vous constatez

⇒ Maintenant faites le merge :

`git merge updates`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git merge updates
Updating 67a08a8..dd0ea54
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

6: sixth Step : Conflict Resolution

- ⇒ Créez une branche avec le nom ‘BAD’

```
git branch BAD
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git branch BAD
```

- ⇒ Modifiez le fichier README.md et ajouter la ligne ‘Trouble’

```
nano README.md
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git checkout BAD
Switched to branch 'BAD'

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (BAD)
$ nano README.md
```

- ⇒ Faites le staging et le commit en une seule ligne

```
git commit -m «commit 6»
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (BAD)
$ git commit -am "commit 6"
[BAD 128c805] commit 6
 1 file changed, 1 insertion(+)
```

- ⇒ Switchez faire la branche principale ‘master’

```
git checkout master
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (BAD)
$ git checkout master
Switched to branch 'master'
```

- ⇒ Maintenant modifiez le fichier README.md et ajoutez la ligne ‘Troubleshooting’

```
nano README.md
```

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ nano README.md
```

- ⇒ Stagging/commiting avec commentaire ‘branche bien faite’

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -am "branche bien faite"
[master accae61] branche bien faite
 1 file changed, 1 insertion(+)
```

- ⇒ Oups !! corriger le commentaire du dernier commit

- ⇒ Affichez l'historique : `git historique`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git historique
* accae61 (HEAD -> master) branche bien faite
* 17ca63f commit 6
| * 128c805 (BAD) commit 6
|/
* dd0ea54 (updates) commit 5
* 67a08a8 commit 4
* 287b06c commit 3
* 59160dc commit 2
* 043b401 commit 1
```

- ⇒ Qu'est-ce que vous constatez ? Expliquez l'objectif de ce petit scenario ?
- ⇒ Faites le merge de la branche 'BAD' : `git merge BAD`

```
youcode@DESKTOP-ORTAJ96 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.
```

- ⇒ Expliquez le message ?
Le message montre qu'il y a des conflits entre 2 fichiers avec le même nom, chacun est modifié dans une branche différente avec des données différentes
- ⇒ Exécutez la commande suivante : `cat README.md` ? Expliquez ?
`cat` nous montre les données sources de conflits avec la branche dans laquelle la modification a été faite

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ cat README.md
#Demo project un simple fichier
modification
Trouble
<<<<< HEAD
Troubleshooting
=====
>>>>> BAD
```

- ⇒ Tapez la commande « `git mergetool` » ? qu'est-ce que vous constatez ?

`git merge tool` nous a affiché 4 écrans à gauche montrant les modifications dans la 1ere branche, à droite montrant la 2eme, vide au centre, et en bas un écran montrant le fichier

7: seventh Step: merge tools

Dans cette section vous êtes amené à installer le logiciel 'p4merge'. Lien d'installation :
<https://www.perforce.com/products/helix-core-apps/merge-diff-tool-p4merge>

- ⇒ Installez p4merge sur votre PC
- ⇒ Sur git bash tapez la commande : `git config --global --list`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config --global --list
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=jihad
user.email=jihad.karamane91@gmail.com
alias.historique=log --oneline --graph --decorate --all
alias.historique2=reflog
merge.tool=p4merge
mergetool.p4merge.path=c:\Program Files\Perforce
```

⇒ Git config --global merge.tool p4merge

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config --global merge.tool p4merge
```

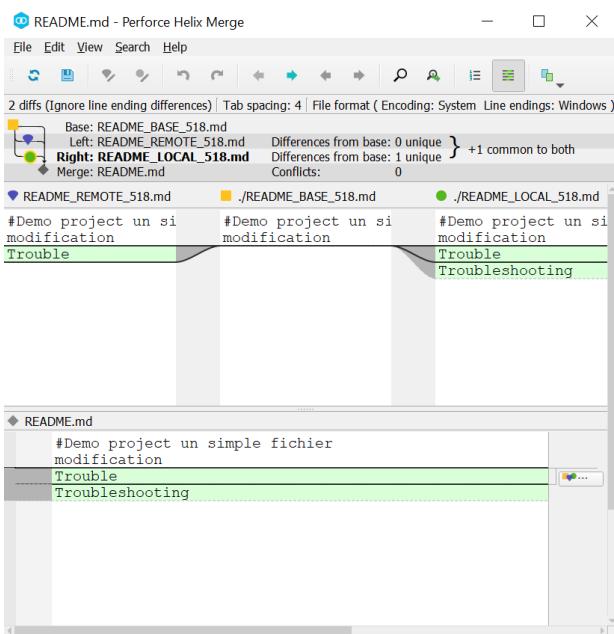
⇒ Git config --global mergetool.p4merge.path "lien d'installation" (.exe)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git config --global mergetool.p4merge.path '/c/Program Files/Perforce/p4merge.exe'
```

⇒ Tapez la commande git mergetool

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ git mergetool
Merging:
README.md

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



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

Meme explication de git mergetool, c'est juste un gui a la place d'un terminal

8: eigth Step: Challenge

Apres avoir résolu les conflits des branches ; vous avez maintenant des fichiers indésirables à rejeter :

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

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ touch .gitignore

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ ls
application.log      README_BACKUP_448.md  README_LOCAL_448.md
licence.txt          README_BASE_290.md   README_REMOTE_290.md
README.md             README_BASE_369.md   README_REMOTE_369.md
README.md.orig        README_BASE_448.md   README_REMOTE_448.md
README_BACKUP_290.md  README_LOCAL_290.md
README_BACKUP_369.md  README_LOCAL_369.md

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master|MERGING)
$ nano .gitignore
```

- ⇒ On laisse que les fichiers : licence.txt Readme.md .gitignore (exemple *.log pour application.log)

```
GNU nano 4.5                               .gitignore
*.log
*.orig
README_*
```

[Read 4 lines]

^G Get Help ^O Write Out ^W Where Is ^K CUT Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

Scénario 2 :

Product BACKLOG 2 :

1 : First Step : Tagging

- ⇒ Déplacez-vous sur la branche Principale :
- ⇒ Créez un TAG avec un nom V1.0 et un commentaire ‘ RELEASE 1.0 ‘ :

git tag v1.0 -m “ RELEASE 1.0 ”

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git tag v1.0 -m "RELEASE 1.0"
```

- ⇒ Affichez les informations sur le TAG

git show v1.0

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git show v1.0
tag v1.0
Tagger: jihad <jihad.karamane91@gmail.com>
Date:   Mon Dec 2 13:16:07 2019 +0100

RELEASE 1.0

commit accae61c00917fdef24504fcfed061a8e361d065 (HEAD -> master, tag: v1.0)
Author: jihad <jihad.karamane91@gmail.com>
Date:   Mon Dec 2 12:03:59 2019 +0100

    branche bien faite

diff --git a/README.md b/README.md
index fe774d3..178db11 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,4 @@
 #Demo project un simple fichier
 modification
 Trouble
+Troubleshooting
```

- ⇒ D'après vous un TAG il sert à faire quoi au juste ?

Un tag sert à afficher les détails d'un commit, l'auteur, la date l'ID du commit et les modifications réalisées.

2 : Second Step : Stashing and Saving work in Progress

- ⇒ Modifiez le fichier README.md , ajoutez une ligne :

nano README.md

- ⇒ 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 ?

git stash

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git stash
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
Saved working directory and index state WIP on master: accae61 branche bien fait
e
```

2. Expliquez le fonctionnement de la commande git Stash

on utilise git stash quand on veut enregistrer l'état actuel du répertoire et l'index , et on veut revenir à un répertoire propre, elle enregistre les modifications locales et reinistialise le répertoire vers le dernier commit.

3. Tapez la commande git stash list

git stash list

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git stash list
stash@{0}: WIP on master: accae61 branche bien faite
```

4. Exécutez la commande git status ?

git status

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md.orig
    README_BACKUP_290.md
    README_BACKUP_369.md
    README_BACKUP_448.md
    README_BASE_290.md
    README_BASE_369.md
    README_BASE_448.md
    README_LOCAL_290.md
    README_LOCAL_369.md
    README_LOCAL_448.md
    README_REMOTE_290.md
    README_REMOTE_369.md
    README_REMOTE_448.md

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

5. Qu'est-ce que vous constatez ?

on constate que le fichier modifié n'apparaît pas sur git status, car les derniers modifications non committés ont été enregistrés ailleurs grâce à la commande git stash.

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

nano Licence.txt

⇒ Faites staging et le commit en une seule ligne :

git commit –am “ stash 1”

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git config --list
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git commit -am "stash 1"
warning: LF will be replaced by CRLF in licence.txt.
The file will have its original line endings in your working directory
[master b94a017] stash 1
  1 file changed, 1 insertion(+)

filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=jihad
user.email=jihad.karamanoglu@gmail.com

⇒ Exécutez la commande « git stash pop » ?
youcode@DESKTOP-ORTAJ96 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: .gitignore
    modified: README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (9f70146aa0019f646fc10226524704fb010fc689)

```

⇒ Qu'est-ce que vous constatez ?

on constate que les modifications enregistrés ailleurs grâce à la git stash reviennent vers la repo locale, et le stash est supprimé de la liste des stash.

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

⇒ Pour cette phase vous devez avoir un compte GITHUB

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

⇒ Créez le remote en https

`git remote add origin https://github.com/jihad19/Sc-narios.git`

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git remote add origin https://github.com/jihad19/sc-narios.git

```

⇒ Examinez le remote

`Git config --list`

⇒ Pushez le tous à travers la commande : `git push --u origin master --tags`

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ git push -u origin master --tags
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 8 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (28/28), 2.53 KiB | 112.00 KiB/s, done.
Total 28 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/jihad19/sc-narios.git
 * [new branch]      master -> master
 * [new tag]          v1.0 -> v1.0
Branch 'master' set up to track remote branch 'master' from 'origin'.

```

- ✓ Expliquez la commande ?

git push -u origin master --tags sert à push le répertoire local et les tags en même temps.

- ✓ Expliquez les options -u et --tags ?

L'option -u indique à git de créer toutes les informations nécessaires pour rendre la branche traçable. Il faut ensuite spécifier le nom du repository distant (par défaut, il se nomme origin) et le nom de la branche.

on utilise --tags lorsque'on a plusieurs tags à push en même temps .

⇒ 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**

mkdir.SSH

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ mkdir .SSH
```

⇒ Déplacez-vous dans le fichier .SSH

Cd .SSH

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ cd .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').

ssh-keygen -t rsa -C 'jihad.karamane91@gmail.com'

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo/.ssh (master)
$ ssh-keygen -t rsa -C "jihad.karamane91@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/users/youcode/.ssh/id_rsa):
/c/users/youcode/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/users/youcode/.ssh/id_rsa.
Your public key has been saved in /c/Users/youcode/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:3Tv0cvQdZ161/W1SFrFDN7ebmWmxuDWSANZKwf2wMkA jihad.karamane91@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
| .E.++ +o |
| ..o.+ . B |
| o ..+ * . |
| = o...o # |
| S + o+.&B |
| . +=*O |
| +.+ B |
| + o |
+---[SHA256]---+

```

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

- ⇒ Sur github créez un autre repo nommé (Monsiteweb)
- ⇒ Ajoutez à l'arborescence toujours sur github le fichier **.gitignore** et un fichier **licence.txt** ‘**APACHE 2.0**’
- ⇒ Déplacez-vous dans le répertoire projets sous GIT

Cd ..

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo/.ssh (master)
$ cd ..

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/demo (master)
$ cd ..

```

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

Git clone « url »

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects
$ git clone https://github.com/jihad19/Monsiteweb.git
Cloning into 'Monsiteweb'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.

```

- ⇒ Vérifiez si le clone est créé.

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'arborescence.
- ⇒ Expliquez ?
- ⇒ Copiez le site télécharger dans votre repo local à travers une seule commande :
Hint : **cp -R ‘~/DESKTOP/initializr/*’ .** (supposons que vous êtes sur votre repolocal)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects
$ cp -r /c/users/youcode/Desktop/initializr /c/Users/youcode/Desktop/projects/Monsiteweb
```

git status

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/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)
    initializr/
nothing added to commit but untracked files present (use "git add" to track)
```

Faites le staging et le commit en une seule ligne **git add . && git commit -m "commit 8"**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git add . && git commit -m "commit 8"
warning: LF will be replaced by CRLF in initializr/initializr/browserconfig.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/initializr/css/main.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/initializr/css/normalize.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/initializr/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/initializr/js/main.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/initializr/js/vendor/moderniz
r-2.8.3-respond-1.4.2.min.js.
The file will have its original line endings in your working directory
[master 0cbda9e] commit 8
 11 files changed, 910 insertions(+)
  create mode 100644 initializr/initializr/apple-touch-icon.png
  create mode 100644 initializr/initializr/browserconfig.xml
  create mode 100644 initializr/initializr/css/main.css
  create mode 100644 initializr/initializr/css/normalize.css
  create mode 100644 initializr/initializr/css/normalize.min.css
  create mode 100644 initializr/initializr/favicon.ico
  create mode 100644 initializr/initializr/index.html
  create mode 100644 initializr/initializr/js/main.js
  create mode 100644 initializr/initializr/js/vendor/modernizr-2.8.3-respond-1.4.
2.min.js
  create mode 100644 initializr/initializr/tile-wide.png
  create mode 100644 initializr/initializr/tile.png
```

⇒ Fait le push à github

Git push -u origin master

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git push -u origin master
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (17/17), 24.54 KiB | 1.23 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0)
To https://github.com/jihad19/Monsiteweb.git
  5599198..0cbda9e master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

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

The screenshot shows a GitHub repository page for 'Monsiteweb / initializr / initializr /'. The top navigation bar includes 'Create new file', 'Upload files', 'Find file', and 'History'. The main area displays a list of files with their commit history:

File	Commit	Time Ago
..	commit 8	15 hours ago
css	commit 8	15 hours ago
js	commit 8	15 hours ago
apple-touch-icon.png	commit 8	15 hours ago
browserconfig.xml	commit 8	15 hours ago
favicon.ico	commit 8	15 hours ago
index.html	Update index.html	15 hours ago
tile-wide.png	commit 8	15 hours ago
tile.png	commit 8	15 hours ago

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 .

The screenshot shows the raw content of the 'index.html' file from the GitHub repository. The file contains the following HTML code:

```
<!DOCTYPE html>
<!--[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]><!--> <html class="no-js" lang=""> <!--<!--> <![endif]-->
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <title>Mon premier Site Web</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="apple-touch-icon" href="apple-touch-icon.png">

    <link rel="stylesheet" href="css/normalize.min.css">
    <link rel="stylesheet" href="css/main.css">

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

- ⇒ Faites le commit sur github
- ⇒ Sur git et sur le repo local , Editez le fichier README.md
`nano README.md`
- ⇒ Faites le staging et le commit en une seule ligne
`git commit -am « commit 9 »`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git commit -am "commit 9"
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
[master e37b846] commit 9
 1 file changed, 2 insertions(+), 1 deletion(-)
```

- ⇒ Qu'est-ce que vous constatez ?
- Trouble shooting :
- ⇒ Executez la commande « git fetch »

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git fetch
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/jihad19/Monsiteweb
  0cbda9e..c3433a7  master      -> origin/master
```

- ⇒ Git status
- ```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/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
```

- ⇒ Expliquez ?
- ⇒ Git pull

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git pull
hint: Waiting for your editor to close the file... 0 [sig] bash 752! s
cket::process: Suppressing signal 18 to win32 process (pid 15292)
 168043 [sig] bash 752! sigpacket::process: Suppressing signal 18 to win32 p
ss (pid 15292)
Merge made by the 'recursive' strategy.
 initializr/initializr/index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- ⇒ Git push
- `git push -u origin master`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ 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), 531 bytes | 265.00 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/jihad19/Monsiteweb.git
 c3433a7..ca6b9a6 master -> master
```

⇒ Verifiez les commit sur github

The screenshot shows a GitHub repository page for 'Monsiteweb'. At the top, there are buttons for 'Branch: master ▾', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download ▾' button. Below this, a list of commits is displayed:

| Author                | Commit Message                                                 | Time Ago                           |
|-----------------------|----------------------------------------------------------------|------------------------------------|
| jihad19               | Merge branch 'master' of https://github.com/jihad18/Monsiteweb | Latest commit ca6b9a6 15 hours ago |
| initializr/initializr | Update index.html                                              | 15 hours ago                       |
| .gitignore            | Create .gitignore                                              | 16 hours ago                       |
| README.md             | commit 9                                                       | 15 hours ago                       |
| licence.txt           | Create licence.txt                                             | 15 hours ago                       |

Below the commits, there is a preview of the 'README.md' file content:

```
Monsiteweb
```

zzz

# Scénario 3 :

## Product BACKLOG :

### 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

A screenshot of a GitHub repository page. The URL is 'Monsiteweb / initializr / initializr / css /'. The branch is 'master'. The commit history shows a single commit by 'jihad19' titled 'Create style.css' made 21 seconds ago. Other files listed include main.css, normalize.css, normalize.min.css, and style.css.

| File              | Commit           | Time Ago       |
|-------------------|------------------|----------------|
| main.css          | commit 8         | 21 hours ago   |
| normalize.css     | commit 8         | 21 hours ago   |
| normalize.min.css | commit 8         | 21 hours ago   |
| style.css         | Create style.css | 21 seconds ago |

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

<h1> modification récente </h1>

A screenshot of a code editor showing a snippet of HTML. Line 18 shows '</head>'. Line 19 shows '<body>'. Line 20 shows '<h1> : <h1> modification récente </h1>'.

```
18 </head>
19 <body>
20 <h1> : <h1> modification récente </h1>
```

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

A screenshot of a GitHub interface showing a file named 'error404.html' with a note 'Rename 404.html to error404.html' and a timestamp '3 minutes ago'.

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

1. Sur gitHub vérifiez la liste des commits

A screenshot of a GitHub commit history for the 'initializr' repository. It shows four commits from 'jihad19' made on December 3, 2019:

- Rename 404.html to error404.html (Verified, c5eb8ea, 1 minute ago)
- Create 404.html (Verified, 3365cccd, 2 minutes ago)
- Update index.html (Verified, 8cee765, 8 minutes ago)
- Create style.css (Verified, 783a8e6, 11 minutes ago)

| Commits on Dec 2, 2019                                         |                         |  |  |
|----------------------------------------------------------------|-------------------------|--|--|
| Merge branch 'master' of https://github.com/jihad18/Monsiteweb | <a href="#">ca6b9a6</a> |  |  |
| jihad19 committed 21 hours ago                                 |                         |  |  |
| commit 9                                                       | <a href="#">e37b846</a> |  |  |
| jihad19 committed 21 hours ago                                 |                         |  |  |
| Update index.html                                              | c3433a7                 |  |  |
| jihad19 committed 21 hours ago                                 |                         |  |  |
| commit 8                                                       | <a href="#">0cbda9e</a> |  |  |
| jihad19 committed 22 hours ago                                 |                         |  |  |
| Create licence.txt                                             | 5599198                 |  |  |
| jihad19 committed 22 hours ago                                 |                         |  |  |
| Create .gitignore                                              | 1738989                 |  |  |
| jihad19 committed 22 hours ago                                 |                         |  |  |
| Initial commit                                                 | 00513a6                 |  |  |
| jihad19 committed 22 hours ago                                 |                         |  |  |

## 2. Sur GITBASH, vérifiez avec « git status »

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (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

[git fetch](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git fetch
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 18 (delta 10), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (18/18), done.
From https://github.com/jihad19/Monsiteweb
 ca6b9a6..c5eb8ea master -> origin/master
```

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

[git pull](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git pull
Updating ca6b9a6..c5eb8ea
Fast-forward
 initializr/initializr/css/style.css | 1 +
 initializr/initializr/error404.html | 1 +
 initializr/initializr/index.html | 1 +
 3 files changed, 3 insertions(+)
 create mode 100644 initializr/initializr/css/style.css
 create mode 100644 initializr/initializr/error404.html
```

## 5. Vérifiez votre repo Local

[Les modifications ont été effectués sur le repo local](#)

| Nom              | Modifié le       | Type                |
|------------------|------------------|---------------------|
| css              | 03/12/2019 11:53 | Dossier de fichiers |
| img              | 02/12/2019 14:07 | Dossier de fichiers |
| js               | 02/12/2019 14:07 | Dossier de fichiers |
| apple-touch-icon | 02/12/2019 14:07 | Fichier PNG         |
| browserconfig    | 02/12/2019 14:07 | Document XML        |
| error404         | 03/12/2019 11:53 | Fichier HTML        |
| favicon          | 02/12/2019 14:07 | Icône               |
| index            | 03/12/2019 11:53 | Fichier HTML        |
| tile             | 02/12/2019 14:07 | Fichier PNG         |
| tile-wide        | 02/12/2019 14:07 | Fichier PNG         |

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

The screenshot shows a GitHub commit page. At the top, it says "1 comment on commit 8cee765". Below that, there's a comment from a user named "jihad19" with the message "commentaire". There are buttons for "Author", "Owner", and "Lock conversation".

7. Sur GITBASH affichez les informations du commit de l'ajout de la balise <h1></h1>  
**git show 8cee765**

The screenshot shows a GitHub commit page for "Update index.html" by "jihad19". It includes a timestamp ("committed 27 minutes ago"), a comment count ("1"), and buttons for "Verified", "Copy", and "8cee765".

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git show 8cee765
commit 8cee7654b58054bb4eae71fd7b298916e7a8049d
Author: jihad19 <57941372+jihad19@users.noreply.github.com>
Date: Tue Dec 3 11:37:35 2019 +0100

 Update index.html

diff --git a/initializr/initializr/index.html b/initializr/initializr/index.html
index debf861..04480b1 100644
--- a/initializr/initializr/index.html
+++ b/initializr/initializr/index.html
....skipping...
commit 8cee7654b58054bb4eae71fd7b298916e7a8049d
Author: jihad19 <57941372+jihad19@users.noreply.github.com>
Date: Tue Dec 3 11:37:35 2019 +0100

 Update index.html

diff --git a/initializr/initializr/index.html b/initializr/initializr/index.html
index debf861..04480b1 100644
--- a/initializr/initializr/index.html
+++ b/initializr/initializr/index.html
@@ -17,6 +17,7 @@
 <script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></scr
ipt>
```

## 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 README.md et faites-le commit (toujours sur GITHUB)



- Sur github et sur le menu ‘branch’ ; Expliquez ce que vous voyez comme message ?  
the branch is even with master, ca veut dire que la branche example contient les memes elements que la branche master

### 2. Repo local :

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

Git checkout –b annulation

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb (master)
$ git checkout -b annulation
Switched to a new branch 'annulation'
```

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

Code index.html

```
17 | <script src="js/vendor/modernizr-2.8.3/respond-1.4.2.min.js"></script>
18 | </head>
19 | <body>
20 | <p class="browserupgrade">You are using an outdated browser. Please <a href="h
21 | <![endif]-->
22 |
```

- Faites-le commit et le staging en une seule ligne.

Git commit –am “commit 10”

- « Git show » pour vérifier vos changements.

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr (annulation)
$ git show
commit b37b44c10f837053be73a20484b30f901897accd (HEAD -> annulation)
Author: jihad <jihad.karamane91@gmail.com>
Date: Tue Dec 3 12:19:04 2019 +0100

 commit 10

diff --git a/initializr/initializr/index.html b/initializr/initializr/index.html
index 04480b1..624f8a7 100644
--- a/initializr/initializr/index.html
+++ b/initializr/initializr/index.html
@@ -17,8 +17,6 @@
 <script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></script>
 </head>
 <body>
- <h1> : <h1> modification récente </h1>
- <!--[if lt IE 8]>
- <p class="browserupgrade">You are using an outdated browser. Please upgrade your browser to improve your experience.</p>
- <![endif]-->

```

### 3. Vérification :

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

`git push origin annulation`

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr (annulation)
$ git push origin annulation
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 390 bytes | 390.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote:
remote: Create a pull request for 'annulation' on GitHub by visiting:
remote: https://github.com/jihad19/Monsiteweb/pull/new/annulation
remote:
To https://github.com/jihad19/Monsiteweb.git
 * [new branch] annulation -> annulation

```

- Maintenant vérifiez votre repo distant.



- Qu'est-ce que vous constatez.

La branche annulation a été ajoutée sur le repo distant avec les modifications

## 3: Third Step: compare pull Requests

- Sur votre repo Distant, un message indiquant qu'une branche a été ajoutée avec un bouton « compare and pull request » , Cliquez sur le bouton ? analysez le rendu de la page ? qu'est-ce que vous constatez ?

Commits on Dec 03, 2019

jihad19 commit 10 b37b44c

Showing 1 changed file with 0 additions and 2 deletions.

Unified Split

initializr/index.html

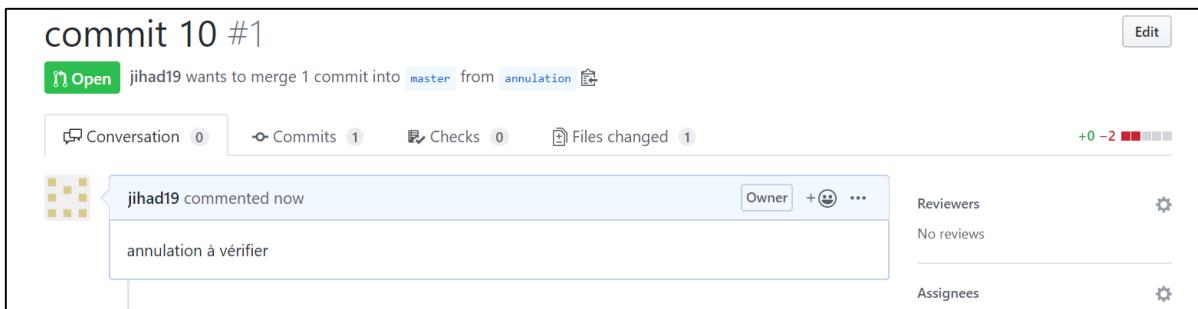
```

@@ -17,8 +17,6 @@
 <script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></script>
</head>
<body>
- <H1> : <h1> modification récente </h1>
- <!--[if lt IE 8]>
+ <p class="browserupgrade">You are using an outdated browser. Please
<![endif]-->

```

les modifications sont colorés en rouge pour la suppression et en vert pour l'ajout pour montrer le changement sur le fichier .

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



- Une page s'ouvre, explorez la page ? qu'est-ce que vous constatez ?
- Surl'onglet ‘conversation’ vous pouvez aussi laisser des commentaires à la personne qui demande 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'

[Git checkout -b ajustement](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr (annulation)
$ git checkout -b ajustement
Switched to a new branch 'ajustement'
```

- on va modifier le fichier 'main.css' (css/main.css) en ajoutant la ligne suivante :

[code main.css](#)

**body {**

**Width :**

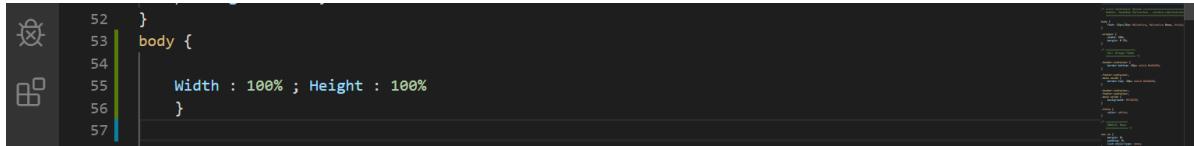
**100% ;**

**Height :**

**100%**

**}**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr (ajustement)
$ code main.css
```



```
52 }
53 body {
54 Width : 100% ; Height : 100%
55
56 }
57 }
```

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

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (ajustement)
$ git push origin ajustement
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 567 bytes | 283.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/jihad19/Monsiteweb.git
 b37b44c..4b800ec 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 ! surely 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?

[git checkout master](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (ajustement)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

4. Maintenant demandez le pull ?

[Git pull](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/jihad19/Monsiteweb
 c5eb8ea..3f34939 master -> origin/master
 * [new branch] Example -> origin/Example
Updating c5eb8ea..3f34939
Fast-forward
 initializr/initializr/index.html | 2 --
 1 file changed, 2 deletions(-)
```

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  
[recursive merge](#)
6. Faites le merge à travers GITBASH  
[git merge ajustement](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git merge ajustement
hint: waiting for your editor to close the file... 1 [sig] bash 958! sigpacket::process: suppressing signal
18 to win32 process (pid 15472)
195184 [sig] bash 958! sigpacket::process: suppressing signal 18 to win32 process (pid 15472)
Merge made by the 'recursive' strategy.
 initializr/initializr/css/main.css | 4 +++
1 file changed, 4 insertions(+)
```

```
MINGW64:/c/Users/youcode/Desktop/projects/Monsiteweb/initializr/initializr/css
Merge branch 'ajustement'
Please enter a commit message to explain why this merge is necessary,
especially if it merges an updated upstream into a topic branch.
#
Lines starting with '#' will be ignored, and an empty message aborts
the commit.
~
```

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

Elle sert à mettre un commentaire justifiant le merge ou d'annuler le merge.

8. Git status

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (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)
 ./main.css

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

9. Si tout est OK ! Pushez les changements?

git push origin master

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git push origin master
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 225 bytes | 225.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/jihad19/Monsiteweb.git
 3f34939..7b280bc 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.

The screenshot shows the GitHub web interface with three main sections:

- Default branch:** Shows a single branch named "master" which was updated 12 minutes ago by "jihad19". There are buttons for "Default" and "Change default branch".
- Your branches:** Shows two branches: "ajustement" and "Example", both of which were deleted just now by "jihad19". Each branch has a "Restore" button.
- Active branches:** Shows the same two deleted branches ("ajustement" and "Example") with their respective "Restore" buttons.

- Sur le local on doit apporter les changements qu'on a fait sur GITHUB :

1. Tapez la commande `git branch -a` ? expliquez le message ?

**Montre toutes les branches locales et distantes**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -a
 ajustement
 annulation
* master
 remotes/origin/Example
 remotes/origin/HEAD -> origin/master
 remotes/origin/ajustement
 remotes/origin/annulation
 remotes/origin/master
```

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

**Git branch -d ajustement**

**Git branch -d annulation**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -d ajustement
Deleted branch ajustement (was 4b800ec).

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -d annulation
Deleted branch annulation (was b37b44c).
```

- Maintenant ! Tapez la commande `git branch -a` ? qu'est-ce que vous constatez ?

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -a
* master
 remotes/origin/Example
 remotes/origin/HEAD -> origin/master
 remotes/origin/ajustement
 remotes/origin/annulation
 remotes/origin/master
```

Les branches locales sont supprimés mais les branches distantes apparaissent toujours en rouge

3. 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 ?

**Git fetch --all --prune**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git fetch --all --prune
Fetching origin
From https://github.com/jihad19/Monsiteweb
 - [deleted] (none) -> origin/Example
 - [deleted] (none) -> origin/ajustement
 - [deleted] (none) -> origin/annulation
```

4. Tapez la commande `git branch -a` ? qu'est-ce que vous constatez ?

**Git branch -a**

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -a
* master
 remotes/origin/HEAD -> origin/master
 remotes/origin/master
```

La liste des branches distantes a été actualisé sur le gitbash.

## 5: fifth Step: The Cleaning up

- Dans cette phase je vous laisse découvrir l'objectif ?
- Sur GITHUB Créez une branche nommée 'updatellicence', éditez le fichier Licence et changer Apache 2.0 par Apache 3.0

A screenshot of a GitHub repository page. The branch is set to 'updatellicence'. The file 'licence.txt' contains the text 'Apache 3.0'. There is one commit by 'jihad19' titled 'Update licence.txt' with a timestamp of 'now'. The commit hash is '7ef5744'. The page includes standard GitHub navigation buttons for 'Raw', 'Blame', 'History', and file operations.

- Sur GITBASH faites un pull global :

[git pull --all](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git pull --all
Fetching origin
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/jihad19/Monsiteweb
 * [new branch] updatellicence -> origin/updatellicence
Already up to date.
```

- Maintenant faites le merge de la branche 'updatellicence'.

[Git merge origin/updatellicence](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git merge origin/updatellicence
Updating 7b280bc..7ef5744
Fast-forward
 licence.txt | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- Executez Git push !

[Git push -u origin master](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 659 bytes | 329.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jihad19/Monsiteweb.git
 7b280bc..7ef5744 master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

- Git branch -d 'updatellicence'

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -d updatellicence
error: branch 'updatellicence' not found.
```

- Exécutez la commande git branch -a ? qu'est-ce que vous constatez sur le remote et aussi sur GTHUB?

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git branch -a
* master
 remotes/origin/HEAD -> origin/master
 remotes/origin/master
 remotes/origin/updatellicence
```



- 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 ? Vérifiez vos Repo (local et distant)

`git push origin --delete updatellicence`

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git push origin --delete updatellicence
To https://github.com/jihad19/Monsiteweb.git
 - [deleted] updatellicence
```

- Pourriez-vous expliquer l'objectif de cette phase ?

Manipuler les branches locales et distantes depuis le git bas

## 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 suivantes :

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

- Et modifiez le titre du site : choisissez le titre que vous voulez

[code index.html](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ code index.html
```

- Faites le staging et le commit en une seule ligne

[git commit –am “commit 12”](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git commit -am "commit 12"
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
 index.html
 ./main.css

nothing added to commit but untracked files present
```

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

[git fetch](#)

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git fetch
```

- Git status ? lisez les commentaires.

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (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)
 index.html
 ./main.css

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

- Maintenant si vous comprenez le commentaire, on a besoin de faire un rebase :

[git pull -- rebase](#)

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

```

youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects/Monsiteweb/initializr/initializr/css (master)
$ git historique
* 7ef5744 (HEAD -> master, origin/master, origin/HEAD) Update licence.txt
* 7b280bc Merge branch 'ajustement'
 \
 * 4b800ec commit 11
 | 3f34939 Merge pull request #1 from jihad19/annulation
 |
 * b37b44c commit 10
 /
* c5eb8ea Rename 404.html to error404.html
* 3365ccd Create 404.html
* 8cee765 Update index.html
* 783a8e6 Create style.css
* ca6b9a6 Merge branch 'master' of https://github.com/jihad18/Monsiteweb
 \
 * c3433a7 Update index.html
 | e37b846 commit 9
 /
* 0cbda9e commit 8
* 5599198 Create licence.txt
* 1738989 Create .gitignore
* 00513a6 Initial commit

```

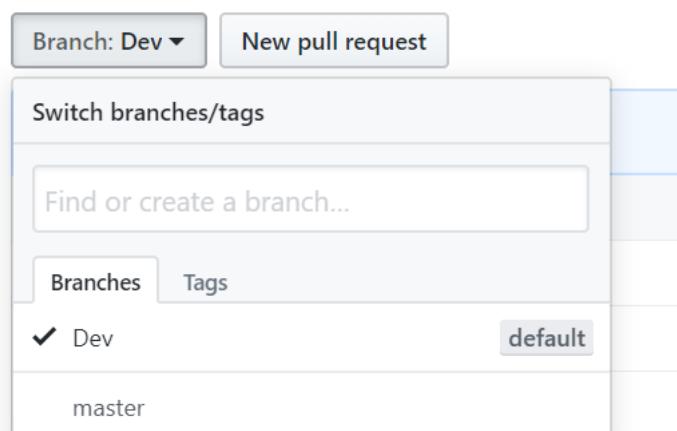
- Vérifiez que tous les changements que vous avez faits sont OK ?
- Expliquez l'utilité du REBASE ?

## 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 : eighth 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

### Rm –r monsiteweb

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects
$ rm -r Monsiteweb
rm: remove write-protected regular file 'Monsiteweb/.git/objects/00/513a60d12427adbd332223d5b9000806842ea2'?
rm: cannot remove 'Monsiteweb/.git/objects/00': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/01/994d0d1ad3cba66ed9c3925320110a25a880b5'?
rm: cannot remove 'Monsiteweb/.git/objects/01': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/02/214d74a0e38d63c8de470c95a2c045318e4fe4'?
rm: cannot remove 'Monsiteweb/.git/objects/02': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/03/0292d2ab7dd11e83f93921e81e7e6699f9a64f'?
rm: cannot remove 'Monsiteweb/.git/objects/03': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/04/480b1173b0082bb60c0f248cd47135a926f316'?
rm: cannot remove 'Monsiteweb/.git/objects/04': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/06/8fa4eeb7e6ca8a61f978ffc6cc6d6ecf6a6cf5'?
rm: cannot remove 'Monsiteweb/.git/objects/06': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/0c/bda9ed4a87877274bd7b433f68691108fb6a1e'?
rm: cannot remove 'Monsiteweb/.git/objects/0c': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/0e/b824b4fc7b62f4abd3c545ff5f73c19c829dd2'?
rm: cannot remove 'Monsiteweb/.git/objects/0e': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/14/326a051ec9d99c4d9edd51b2b2f7006701976a'?
rm: cannot remove 'Monsiteweb/.git/objects/14': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/16/d589176339d7c26c262e36e8c21293b4251cdd'?
rm: cannot remove 'Monsiteweb/.git/objects/16': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/17/3898986083bab936164abd39c87d593c4a6d9f'?
rm: remove write-protected regular file 'Monsiteweb/.git/objects/17/a1d8b8e8779b535fddc1a2d51371c235c40235'?
rm: cannot remove 'Monsiteweb/.git/objects/17': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/1a/d5104cad4a1035396af1d62c2f4ece985687dd'?
rm: cannot remove 'Monsiteweb/.git/objects/1a': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/1c/ed46ed89340c83cf93f45dc6f5da5f88c73f4a'?
rm: cannot remove 'Monsiteweb/.git/objects/1c': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/1e/b20ad3b635dad45afd3875c156b0b37b3071b1'?
rm: cannot remove 'Monsiteweb/.git/objects/1e': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/20/d10c6d3f673baaa1830afdf19fe1eb5f7f27d3c2'?
rm: cannot remove 'Monsiteweb/.git/objects/20': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/22/351e00c724f88869a26f0d5189c6db5f5209ff'?
rm: cannot remove 'Monsiteweb/.git/objects/22': Directory not empty
rm: remove write-protected regular file 'Monsiteweb/.git/objects/29/0464b3fdbb9d6409acf2630c788bb8b3980f19'?
rm: remove write-protected regular file 'Monsiteweb/.git/objects/29/bd70b2cee7cd0639dd9ef75ff54633c7a791f4'?
rm: cannot remove 'Monsiteweb/.git/objects/29': Directory not empty
```

- Maintenant Clonez via https ou SSH si vous l'avez réussi

Git clone <https://github.com/jihad19/Monsiteweb.git>

```
youcode@DESKTOP-ORTAJ96 MINGW64 ~/Desktop/projects
$ git clone https://github.com/jihad19/Monsiteweb.git
cloning into 'Monsiteweb'...
remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 73 (delta 26), reused 66 (delta 23), pack-reused 0
Unpacking objects: 100% (73/73), done.
```

- Vérifiez si le contenu est récupéré

Tout le contenu a été récupéré

- Vérifiez les branches et ou pointe le pointeur : HEAD

Head pointe vers Dev

- Maintenant on a besoin de la branche master comme base :  
[git checkout master](#)
- Toutefois la branche par défaut est : [Dev](#)



Quick Challenge: Try and Catch ()

On va imaginer le mini Scénario suivant :

- Sur GitHub, on va éditez le fichier Readme.md
- Localemement on est sur la branche DEV.
- On a fait des changements sur le REPO distant, surement on a besoin d'un fetch  
[git fetch](#)

Mot clés :

Git fetch, git push -u origin master, git pull, git branch -a; git branch -d ; git push origin :option ;