RAPPORT DE BRIEF

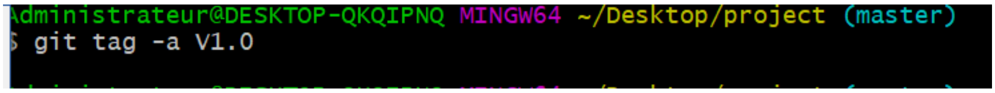
SINARIO 2

29 NOVEMBRE 2019

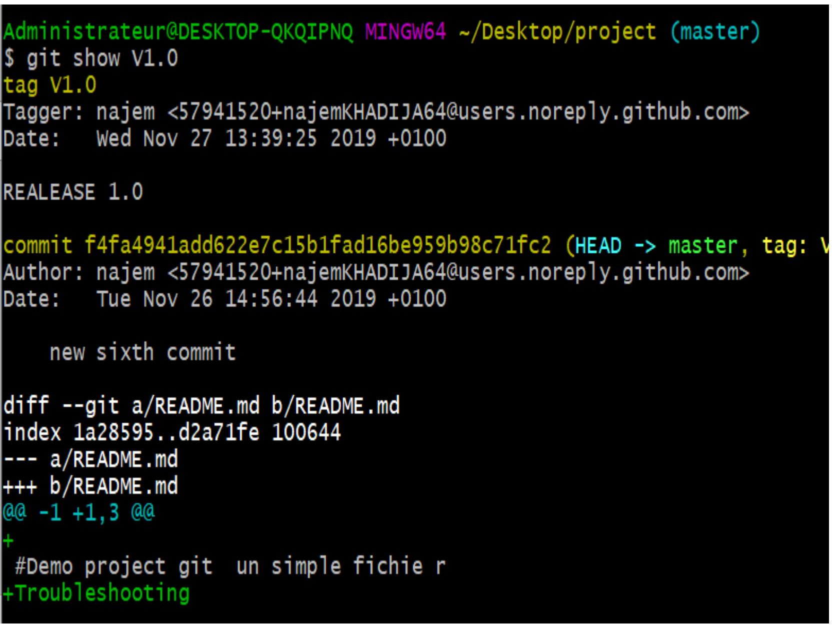
YOUCODE

REALISE PAR :JAAFAR M'HAMAED , KHADIJA NAJEM,MERIEM ELMEDIOUNY, IMAD ELBERDAOUI , GUERMOUD MARIA

First Step : Tagging



```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git tag -a V1.0
```



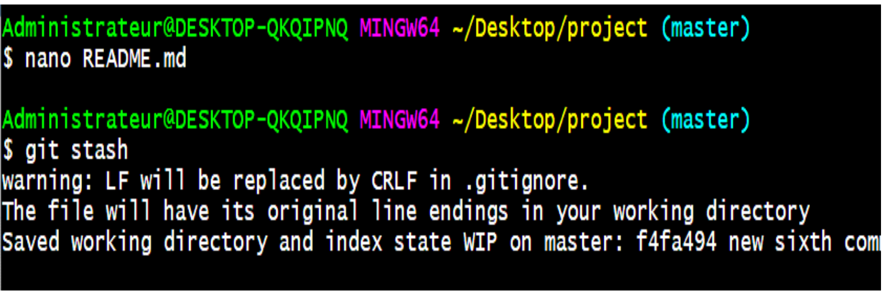
```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git show V1.0
tag V1.0
Tagger: najem <57941520+najemKHADIJA64@users.noreply.github.com>
Date:   Wed Nov 27 13:39:25 2019 +0100

REALEASE 1.0

commit f4fa4941add622e7c15b1fad16be959b98c71fc2 (HEAD -> master, tag: V
Author: najem <57941520+najemKHADIJA64@users.noreply.github.com>
Date:   Tue Nov 26 14:56:44 2019 +0100

    new sixth commit

diff --git a/README.md b/README.md
index 1a28595..d2a71fe 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,3 @@
+
 #Demo project git  un simple fichie r
+Troubleshooting
```

Git has the ability to tag specific points in a repository’s history as being important. Typically, people use this functionality to mark release points (`v1.0`, `v2.0` and so on). In this section, you’ll learn how to list existing tags, how to create and delete tags, and what the different types of tags are.

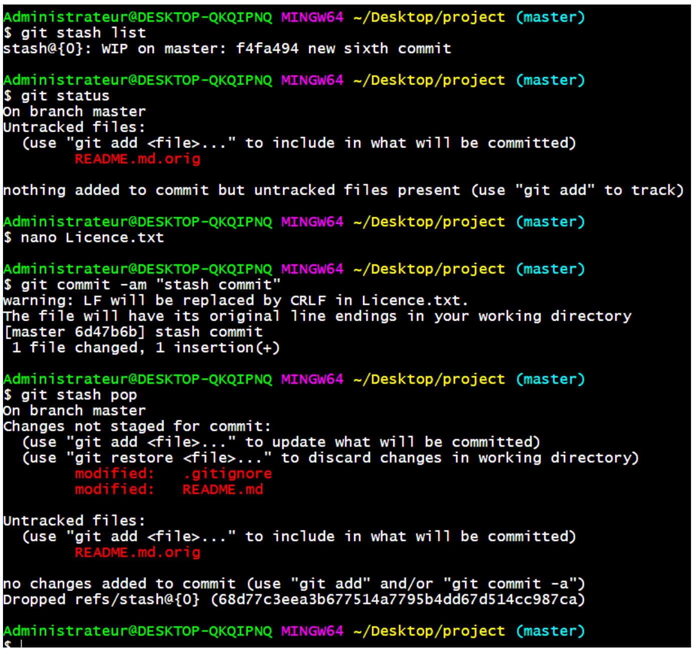
Git show v1.0 : That shows the tagger information, the date the commit was tagged, and the annotation message before showing the commit information.

Second Step : Stashing and Saving work in Progress

git stash : Use git stash when you want to record the current state of the working directory and the index, but want to go back to a clean working directory be listed .



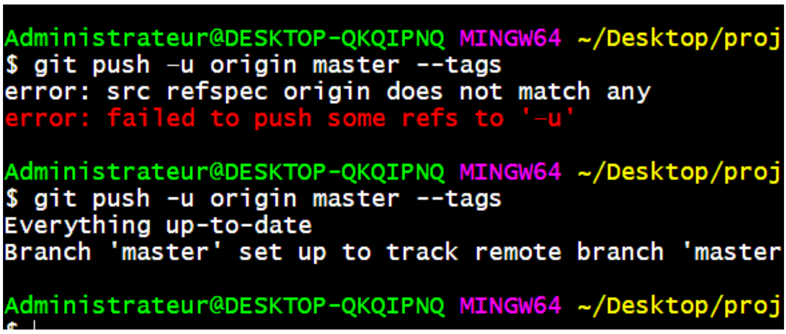
```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ nano README.md

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (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: f4fa494 new sixth com
```



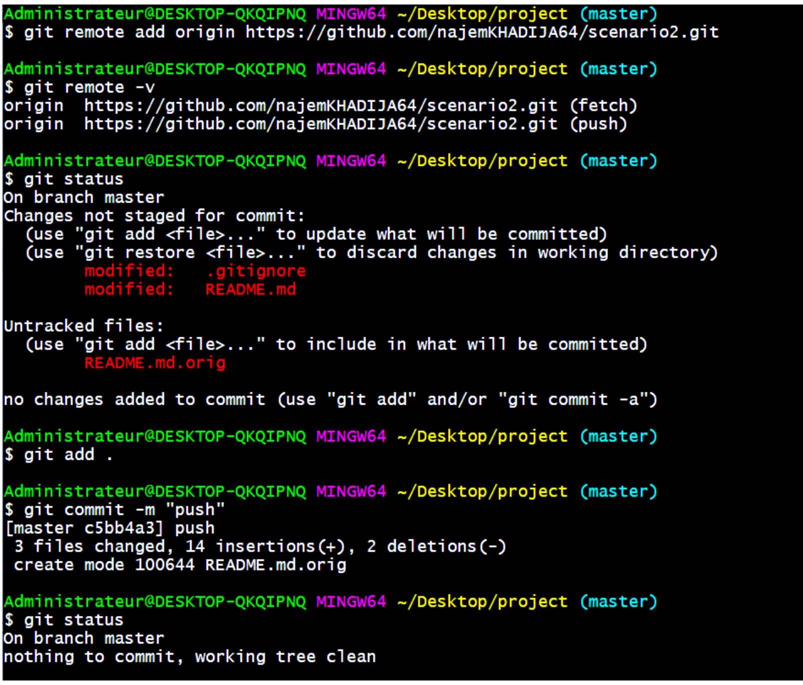
```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git stash list
stash@{0}: WIP on master: f4fa494 new sixth commit

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md.orig

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

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ nano Licence.txt

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git commit -am "stash commit"
warning: LF will be replaced by CRLF in Licence.txt.
The file will have its original line endings in your working directory
[master 6d47b6b] stash commit
 1 file changed, 1 insertion(+)

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md.orig

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (68d77c3eea3b677514a7795b4dd67d514cc987ca)

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$
```

Git stash liste : **The modifications stashed away by this command can be listed with**

Third step: Voyage sur Github, Local Repo to Github Repo



```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/proj
$ git push -u origin master --tags
error: src refspec origin does not match any
error: failed to push some refs to '-u'

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/proj
$ git push -u origin master --tags
Everything up-to-date
Branch 'master' set up to track remote branch 'master

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/proj
$ |
```



```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git remote add origin https://github.com/najemKHADIJA64/scenario2.git

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git remote -v
origin  https://github.com/najemKHADIJA64/scenario2.git (fetch)
origin  https://github.com/najemKHADIJA64/scenario2.git (push)

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git status
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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md.orig

no changes added to commit (use "git add" and/or "git commit -a")

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git add .

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git commit -m "push"
[master c5bb4a3] push
 3 files changed, 14 insertions(+), 2 deletions(-)
 create mode 100644 README.md.orig

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

What is git remote add origin ?

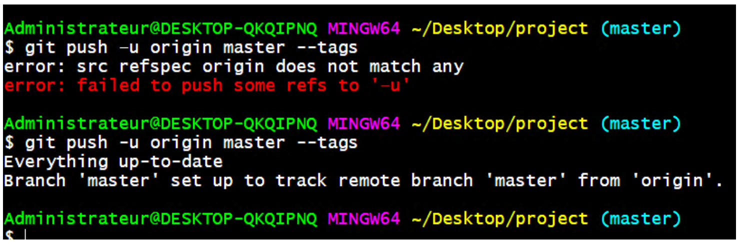
It is communicate with the outside world, git uses what are called remotes. These are repositories other than the one on your local disk which you can push your changes into (so that other people can see them) or pull from (so that you can get others changes).

Fourth Step : Mini challenge (optionnel)

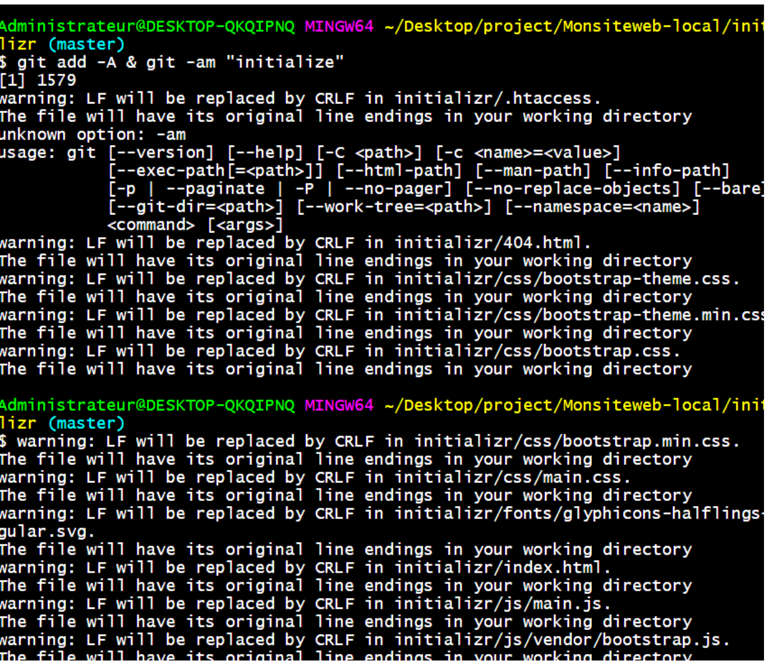
Fifth Step: Création d'une local copy

Sur Github création un autre repo nommé (Monsiteweb).

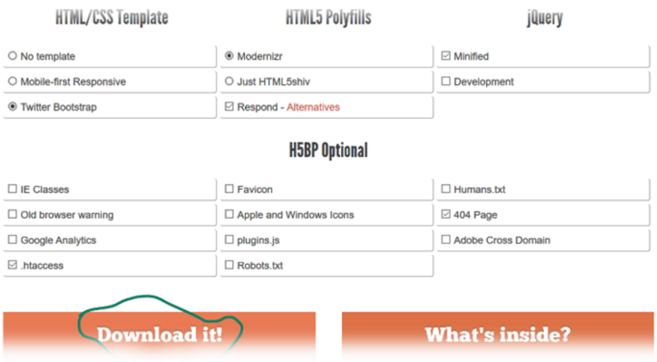
Sixth step: Sending the website



```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git push -u origin master --tags
error: src refspec origin does not match any
error: failed to push some refs to '-u'

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git push -u origin master --tags
Everything up-to-date
Branch 'master' set up to track remote branch 'master' from 'origin'.

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ |
```



```
Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project/Monsiteweb-local/init
lizr (master)
$ git add -A & git -am "initialize"
[1] 1579
warning: LF will be replaced by CRLF in initializr/.htaccess.
The file will have its original line endings in your working directory
unknown option: -am
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]
warning: LF will be replaced by CRLF in initializr/404.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/css/bootstrap-theme.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/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 initializr/css/bootstrap.css.
The file will have its original line endings in your working directory

Administrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project/Monsiteweb-local/init
lizr (master)
$ warning: LF will be replaced by CRLF in initializr/css/bootstrap.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in 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/fonts/glyphicons-halflings-
gular.svg.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in initializr/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in 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/js/vendor/bootstrap.js.
The file will have its original line endings in your working directory
```

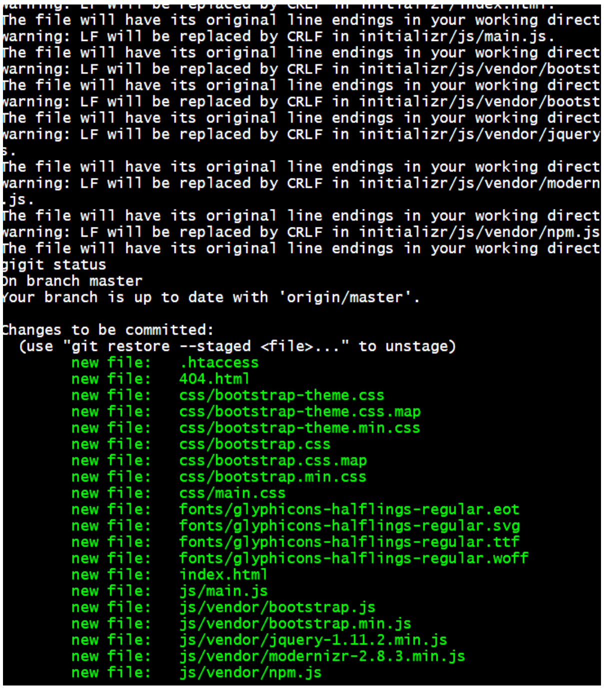


HTML/CSS Template
○ No template
○ Mobile-first Responsive
● Twitter Bootstrap

HTML5 Polyfills
● Modernizr
○ Just HTML5shiv
☑ Respond - Alternatives

jQuery
☑ Minified
☐ Development

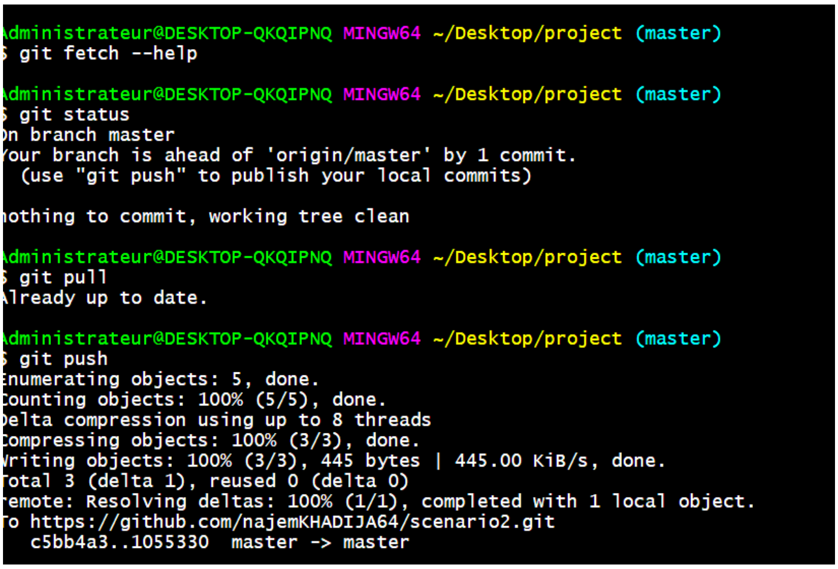
H5BP Optional
☐ IE Classes
☐ Old browser warning
☐ Google Analytics
☑ .htaccess
☐ Favicon
☐ Apple and Windows Icons
☐ plugins.js
☐ Robots.txt
☐ Humans.txt
☑ 404 Page
☐ Adobe Cross Domain



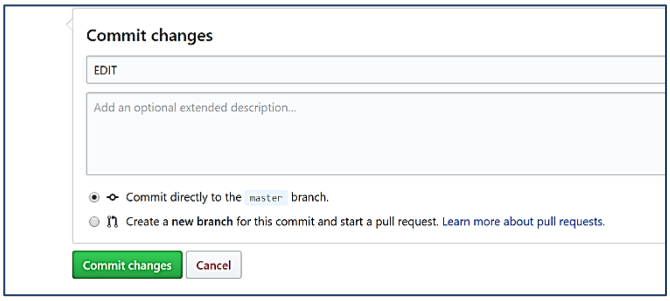
```
warning: LF will be replaced by CRLF in initializr/index.html.
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/main.js.
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/vendor/bootst
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/vendor/bootst
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/vendor/jquery
s.
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/vendor/modern
.js.
The file will have its original line endings in your working direct
warning: LF will be replaced by CRLF in initializr/js/vendor/npm.js
The file will have its original line endings in your working direct
gigit status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .htaccess
        new file:   404.html
        new file:   css/bootstrap-theme.css
        new file:   css/bootstrap-theme.css.map
        new file:   css/bootstrap-theme.min.css
        new file:   css/bootstrap.css
        new file:   css/bootstrap.css.map
        new file:   css/bootstrap.min.css
        new file:   css/main.css
        new file:   fonts/glyphicons-halflings-regular.eot
        new file:   fonts/glyphicons-halflings-regular.svg
        new file:   fonts/glyphicons-halflings-regular.ttf
        new file:   fonts/glyphicons-halflings-regular.woff
        new file:   index.html
        new file:   js/main.js
        new file:   js/vendor/bootstrap.js
        new file:   js/vendor/bootstrap.min.js
        new file:   js/vendor/jquery-1.11.2.min.js
        new file:   js/vendor/modernizr-2.8.3.min.js
        new file:   js/vendor/npm.js
```

Après le téléchargement du initializr, on constate qu'on a des différents dossiers et fichiers. Parmi les , on a : un dossier Css pour styler un site , un dossier de fonts pour varier les types du police et un fichier de HTML .

Seventh step : Fetch and pull



```
dministrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git fetch --help

dministrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git status
n branch master
our branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

othing to commit, working tree clean

dministrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git pull
lready up to date.

dministrateur@DESKTOP-QKQIPNQ MINGW64 ~/Desktop/project (master)
$ git push
numerating objects: 5, done.
ounting objects: 100% (5/5), done.
elta compression using up to 8 threads
ompressing objects: 100% (3/3), done.
riting objects: 100% (3/3), 445 bytes | 445.00 KiB/s, done.
otal 3 (delta 1), reused 0 (delta 0)
emote: Resolving deltas: 100% (1/1), completed with 1 local object.
o https://github.com/najemKHADIJA64/scenario2.git
   c5bb4a3..1055330  master -> master
```



Commit changes

EDIT

Add an optional extended description...

● Commit directly to the `master` branch.

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

The git fetch command downloads commits, files, and refs from a remote repository into your local repo. Fetching is what you do when you want to see what everybody else has been working on. It's similar to svn update in that it lets you see how the central history has progressed, but it doesn't force you to actually merge the changes into your repository.

On constate que les modifications et le commit qu'on a fait sur Github sont affichés aussi sur Git .