

Introduction to Version Control

Assignment

1. Git Setup

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment
$ git --version
git version 2.20.1.windows.1

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment
$ |
```

2. Initialize a Git Repository

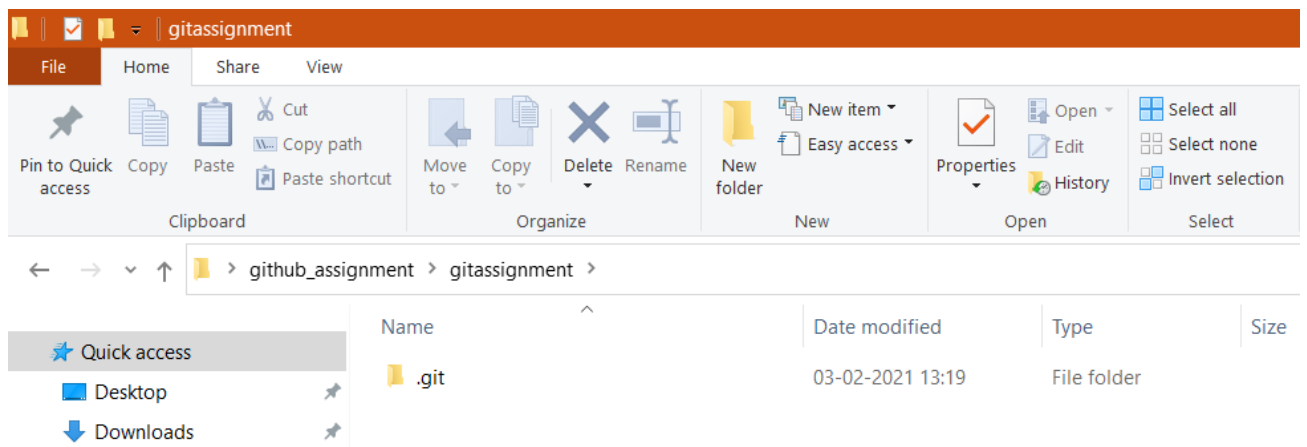
- ⇒ Open git bash inside the project folder (gitassignment is the name of my project folder)
- ⇒ git init

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment
$ git --version
git version 2.20.1.windows.1

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment
$ git init
Initialized empty Git repository in C:/Users/LENOVO/Desktop/github_assignment/gitassignment/.git/

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$
```



3. Add files to the repository

⇒ Using touch filename create files

⇒ Git add .

```
MINGW64:/c:/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file1.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file2.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file3.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

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

        file1.txt
        file2.txt
        file3.txt

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

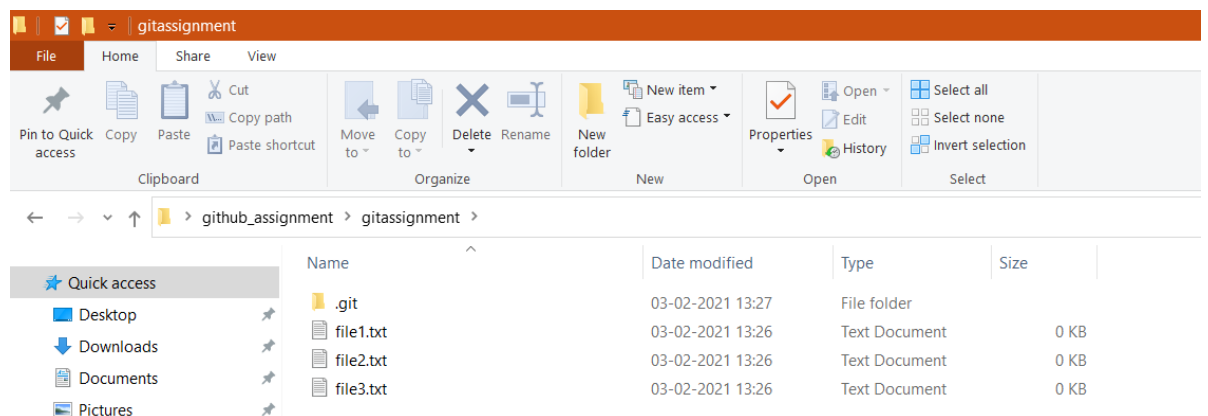
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
```



4. Unstage 1 file

⇒ **Git rm --cached filename**

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git rm --cached file3.txt
rm 'file3.txt'

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1.txt
        new file:   file2.txt

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

        file3.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

5. Commit the file

⇒ **Git commit -m “message”**

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1.txt
        new file:   file2.txt

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

        file3.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git commit -m "Initial commit"
[master (root-commit) a759349] Initial commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt
 create mode 100644 file2.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$
```

6. Add a remote

- ⇒ **Create a repository on github and the copy the https url**
- ⇒ **In gitbash write**
- ⇒ **Git remote add origin <url>**
- ⇒ **<https://github.com/damineesaini/MyRepository> (link of my git repository)**

```
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git remote add origin https://github.com/damineesaini/MyRepository.git

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git remote -v
origin  https://github.com/damineesaini/MyRepository.git (fetch)
origin  https://github.com/damineesaini/MyRepository.git (push)

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$
```

7. Undo changes to a particular file

⇒ **Git checkout -- <filename>**

```
MINGW64/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ vi file2.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file2.txt

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

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git checkout -- file2.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master
nothing to commit, working tree clean

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

8. Push changes to Github

⇒ **Git push origin master**

```
MINGW64/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 217 bytes | 54.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/damineesaini/MyRepository.git
 * [new branch]      master -> master

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$
```

9. Clone the repository

- ⇒ Git clone <url of repository from github>
- ⇒ Cloned the repository (MyRepository) in gitclone folder

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitclone
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$ git clone https://github.com/damineesaini/MyRepository.git
Cloning into 'MyRepository'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$ |
```

10. Add changes to one of the copies and pull the changes in the other.

- ⇒ Gitclone is the folder where I cloned my github repository (MyRepository). I did some changes in file1 here and pushed them

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitclone/MyRepository
c
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$ cd MyRepository/
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git add .
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git commit -m "made some changes"
[master d649980] made some changes
1 file changed, 2 insertions(+)
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 41.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/damineesaini/MyRepository.git
a759349..d649980 master -> master
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ |
```

- ⇒ Gitassignment is my main project folder which I created initially.
- ⇒ Pulled changes in it using => git pull origin master

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/damineesaini/MyRepository
 * branch      master      -> FETCH_HEAD
   a759349..d649980 master  -> origin/master
Updating a759349..d649980
Fast-forward
 file1.txt | 2 ++
 1 file changed, 2 insertions(+)

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

11. Check differences between a file and its staged version

- ⇒ Git diff

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ vi file2.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git diff
warning: LF will be replaced by CRLF in file2.txt.
The file will have its original line endings in your working directory
diff --git a/file2.txt b/file2.txt
index e69de29..b9f36e0 100644
--- a/file2.txt
+++ b/file2.txt
@@ -0,0 +1 @@
+trying to change content

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .
warning: LF will be replaced by CRLF in file2.txt.
The file will have its original line endings in your working directory

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git commit -m "changes"
[master 8d56d12] changes
 2 files changed, 1 insertion(+)
 create mode 100644 .file5.txt.swp

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

12. Ignore a few files to be checked in

- ⇒ Created .gitignore and mentioned the filename in it that you want to be ignored.
- ⇒ Then commit

```
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch trial.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch .gitignore

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ vi .gitignore

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.txt

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

        .gitignore
        file3.txt

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

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git commit -m "adding gitignore"
[master bb56c66] adding gitignore
3 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 .gitignore
create mode 100644 file3.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/qithub_assignment/gitassignment (master)
```


13. Create a new branch.

- ⇒ **Git checkout -b <branchname> master**
- ⇒ **And also added some files to the new branch**

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git checkout -b firstbranch master
Switched to a new branch 'firstbranch'

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ touch branchfile.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git add .

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git commit -m "adding a branch"
[firstbranch 27a3c60] adding a branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 branchfile.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ touch file5.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git add .

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git commit -m "adding another file"
[firstbranch f82d8af] adding another file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file5.txt
```

14. Diverge them with commits

- ⇒ **Git checkout master**
- ⇒ **Git merge firstbranch**

```
LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git checkout master
Switched to branch 'master'

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git merge firstbranch
Updating 12bf8f2..f82d8af
Fast-forward
 branchfile.txt | 0
  file5.txt      | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 branchfile.txt
 create mode 100644 file5.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

15. Edit the same file at the same line on both branches and commit

⇒ Switched to newbranch and edited file5 and commit change

⇒ Switch to master edit file5 and commit change

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git checkout firstbranch
Switched to branch 'firstbranch'

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ vi file5.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git add .
warning: LF will be replaced by CRLF in file5.txt.
The file will have its original line endings in your working directory

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git commit -m "from firstbranch"
[firstbranch 5c158d3] from firstbranch
1 file changed, 2 insertions(+)

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (firstbranch)
$ git checkout master
Switched to branch 'master'

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ vi file5.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git commit -m "from master"
[master 595488b] from master
1 file changed, 1 insertion(+), 1 deletion(-)
```

16. Try merging and resolve merge conflicts

- ⇒ Try to merge new branch created with master
- ⇒ Conflict is shown and merge is failed
- ⇒ The edit the file5 choose to keep the correct content and then add and commit

```
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git merge firstbranch
Auto-merging file5.txt
CONFLICT (content): Merge conflict in file5.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ vi file5.txt

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git merge firstbranch
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git add .

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git commit -m "after correcting conflict"
[master 6f4244b] after correcting conflict

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git merge firstbranch
Already up to date.

LENOVO@LAPTOP-UR011HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ |
```

Below is the file5.txt with merge conflict

```
<<<<<< HEAD
editing i the jshlkdjsah
```

```
=====
trying to edit it in firstbranch
>>>>>> firstbranch
```

[illegible]

17. Stash the changes and pop them

⇒ Git stash

⇒ Git pop

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ vi file3.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file3.txt

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

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git stash
warning: LF will be replaced by CRLF in file3.txt.
The file will have its original line endings in your working directory
Saved working directory and index state WIP on master: 6f4244b after correcting conflict

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file3.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (11516cef47536113a8715d3dfa62307872b7b738)

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .


LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git commit -m "last one"
[master 0621b22] last one
 1 file changed, 2 insertions(+)
```

18. Add the following code to your .bashrc file : color_prompt="yes"

```
parse_git_branch() {  
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(1)/'  
}  
  
if [ "$color_prompt" = yes ]; then  
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\  
$(parse_git_branch)\[\033[00m\]\$ '  
else  
PS1='\u@\h:\W $(parse_git_branch)\$ '  
fi  
unset color_prompt force_color_prompt
```

- ⇒ In windows gitbash bashrc file was created initially.
- ⇒ And that's why it showed warning after reopening the gitbash and then bashrc_profile was reated automatically

[illegible]


```
 /usr/bin/bash --login -i
```

```
WARNING: Found ~/.bashrc but no ~/.bash_profile, ~/.bash_login or ~/.profile.
```

```
This looks like an incorrect setup.
```

```
A ~/.bash_profile that loads ~/.bashrc will be created for you.
```

```
LENOVO@LAPTOP-URO11HCJ:MyRepository (master)$ |
```

```
 /usr/bin/bash --login -i
```

```
LENOVO@LAPTOP-URO11HCJ:MyRepository (master)$
```

Bashrc file in Ubuntu (linux)

```
daminee@LAPTOP-URO11HCJ: ~
color_prompt="yes"
parse_git_branch() {
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*)/(\\1)/'
}
if [ "$color_prompt" = yes ]; then
    PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\033[00m\]\$ '
else
    PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
-- INSERT --
```