

# Getting Started with Git

I will be demonstrating using Windows PowerShell however step would not be different while using Git Bash or Command Prompt

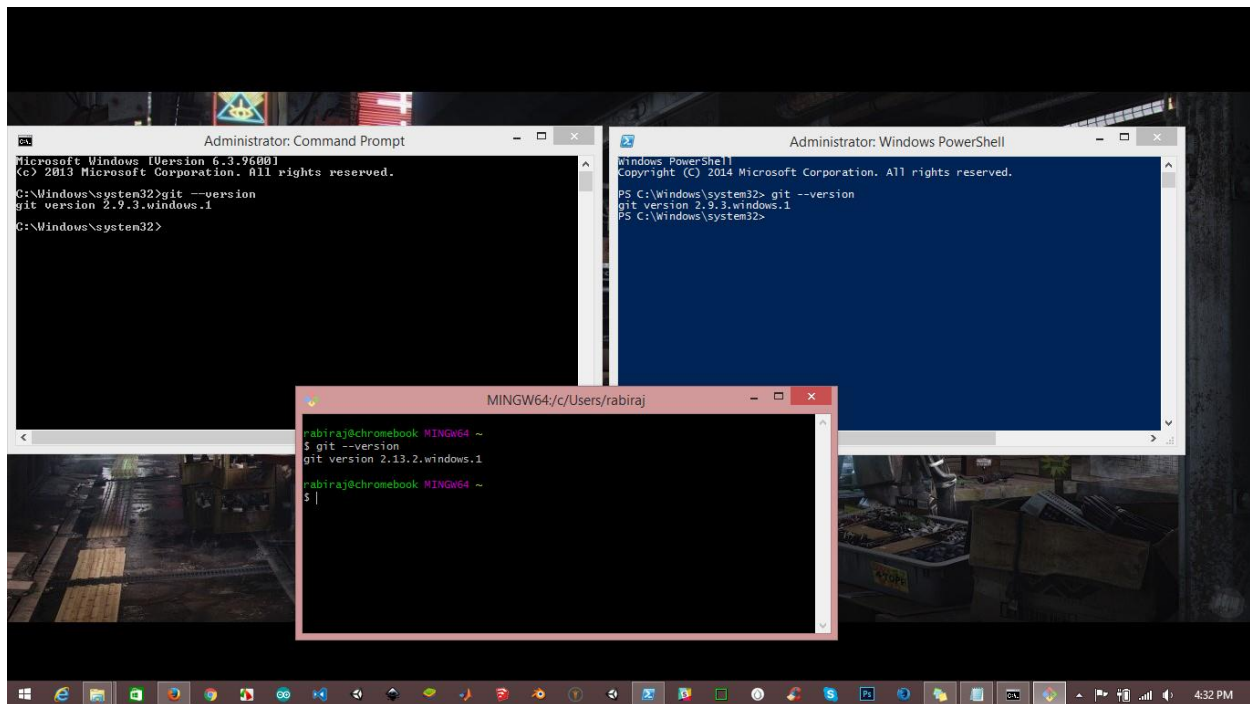
## 1. Checking Version

Git could be opened from git bash, command prompt, powershell or git gui after installation of git client from <http://git-scm.com>

Optionally we could use a number of git gui with separate installation

Few of them includes:

- SourceTree [Free- Mac, Windows],
- Github Desktop [Free- Mac, Windows],
- GitKraken [Free for non-commercial use- Linux, Mac, Windows],
- SmartGit [Free for non-commercial use- Linux, Mac, Windows]

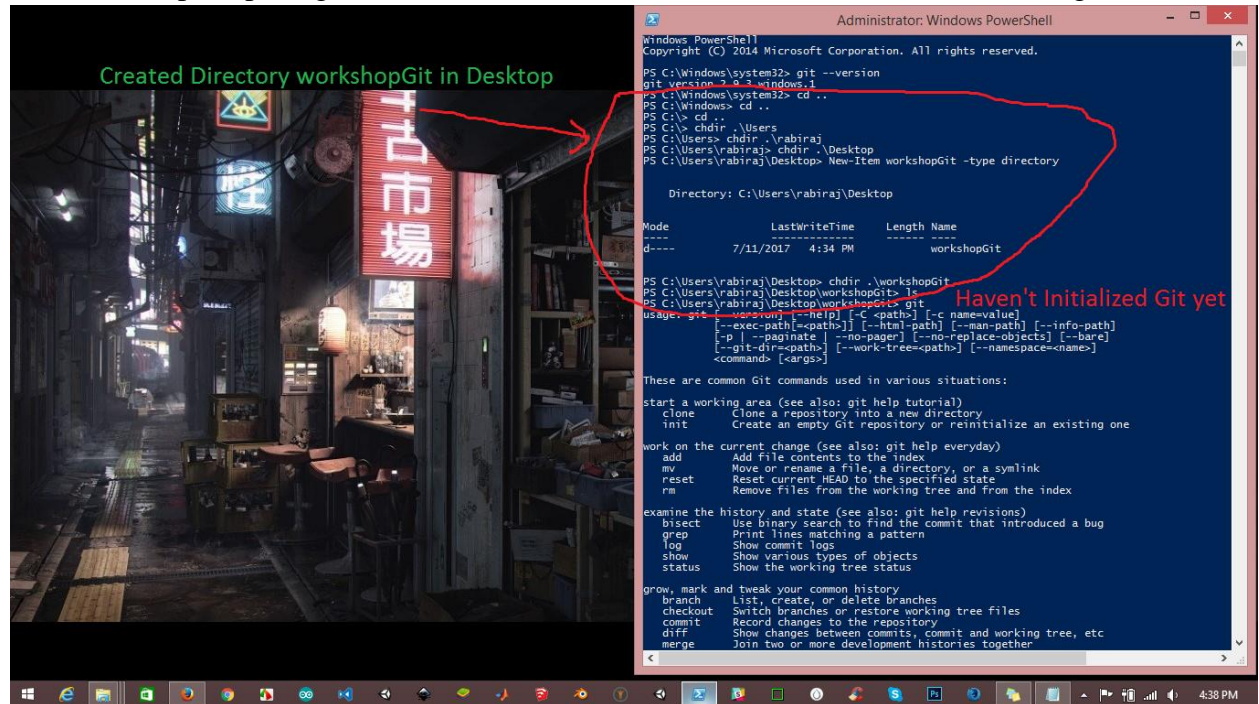


To Check version we should type command *git --version*

## 2. Traversing through directory and creating Directory in Desktop

I have used *chdir* to change the directory and *New-Item* to create new directory and files in PowerShell.

In command prompt or git bash we can use *cd* and *mkdir* to achieve the same thing.

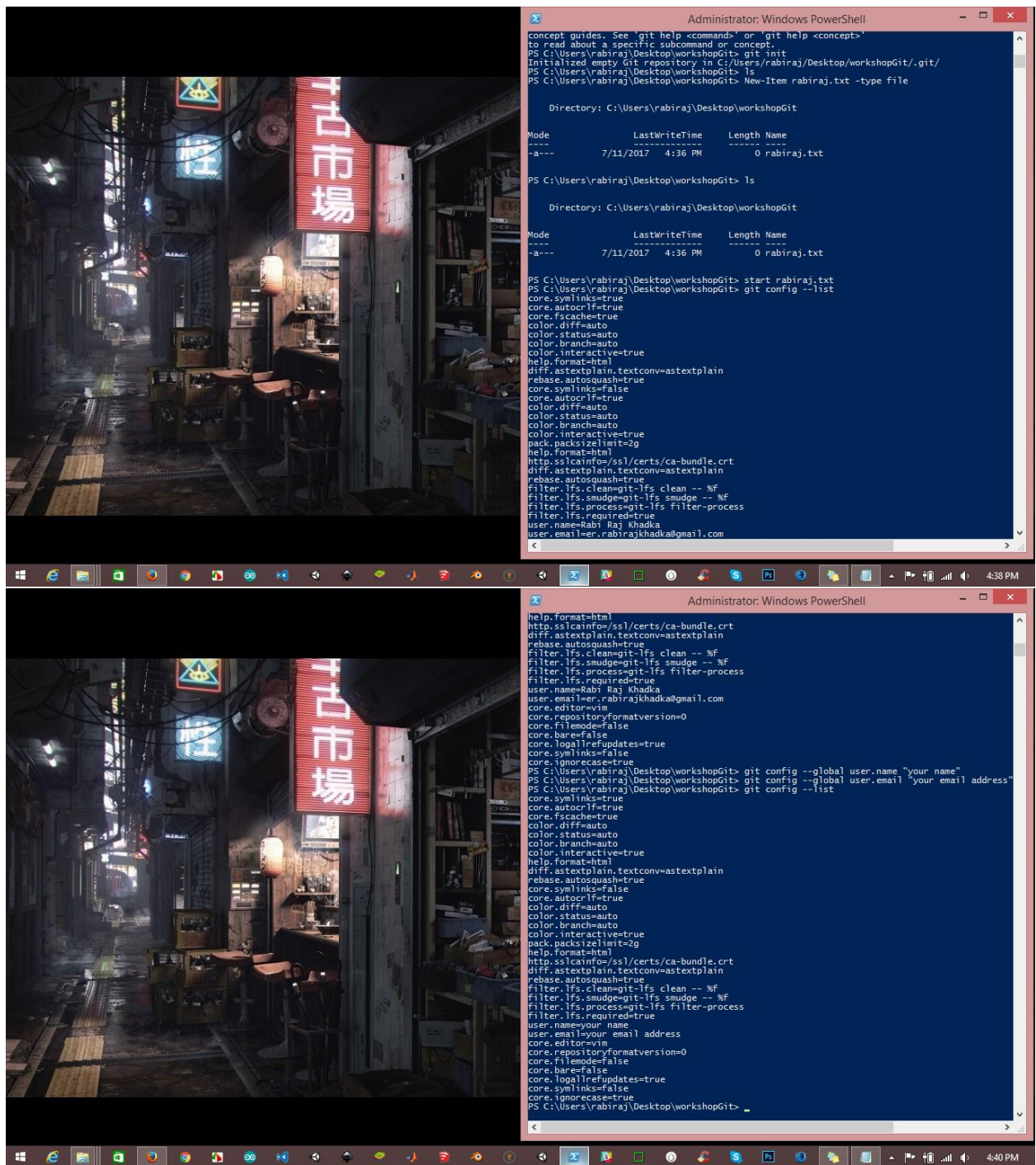


## 3. Git Configuration

Before using Git in Local environment and in the newly created folder workshopGit we have to initialize the git which we haven't initialized yet

To initialize git enter command *git init* which will create folder named .git in workshopGit.

To configure Git Client first let's see the configurable attributes by entering command *git config --list*



Here are the list of configurable attributes however we don't need to configure everything we should configure user.name and user.email which will help to create the information about who committed the files in repository.

Don't Worry if you are not able to see user.name and user.email field while entering above commands.

Lets configure user.name and user.email by entering following commands

*git config --global user.name "your name"*  
*git config --global user.email "your email address"*

If you are getting **multiple values** on user.name and user.email just use --replace-all option as below

*git config --global --replace-all user.name "your name"*  
*git config --global --replace-all user.email "your email address"*

#### 4. Add and Commit

After configuring the git and initializing repository let's make new files using *New-Item rabiraj.txt -type file* *replace rabiraj.txt with filename of your choice*

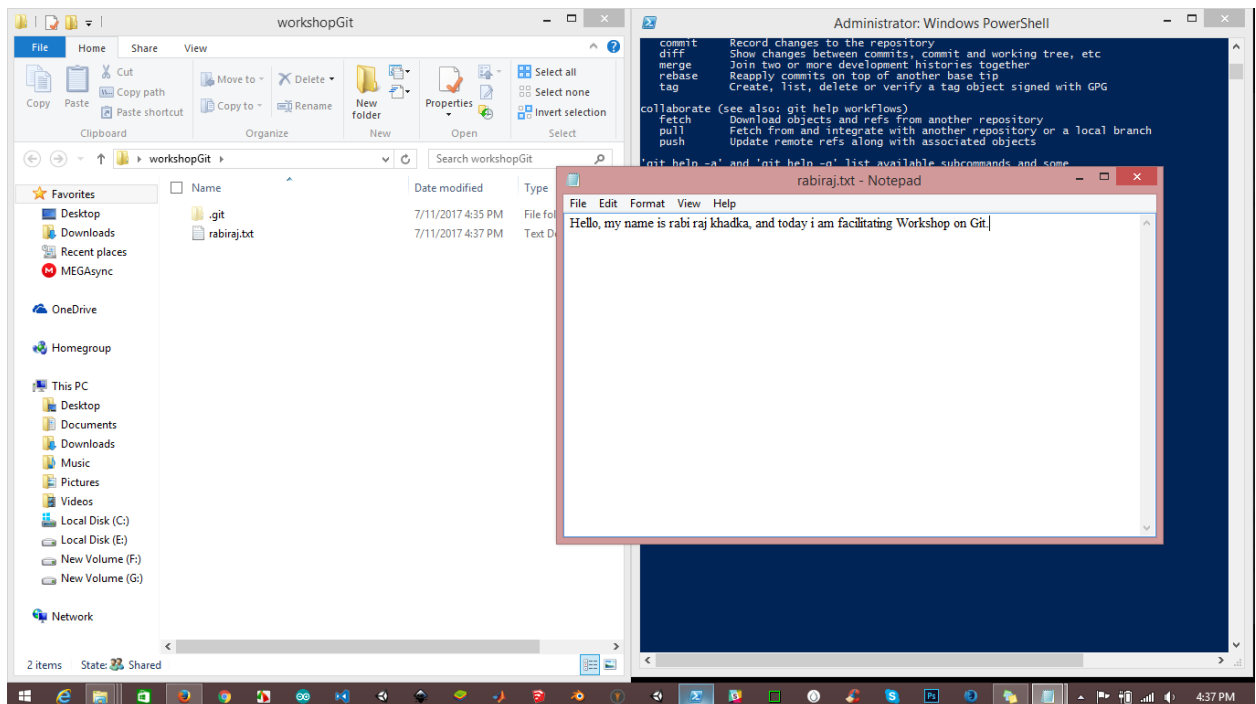
Optionally you could create files by entering following command  
*touch rabiraj.txt*

*echo "contents of file ">rabiraj.txt*

After creating file lets edit some contents of it

*start rabiraj.txt* [opens default txt opening application] *or*  
*vim rabiraj.txt* [(if using git bash)opens vim editor]

to exit from vim press ESC and then press : and wq and <ENTER> here w is for write and q is for quit



or simply :w to write and stay within vim or :q to exit to git bash.



To add files into staging area or index use

*git add <filename>* for individual file or  
*git add .* to add all file and folder

To commit files to repository enter

*git commit -m "message of commit"*

To add tag we use

*git tag vx.x -a -m "message for tag"* x.x is version name here

To check status of tracked, untracked or indexed file we could use

*git status*

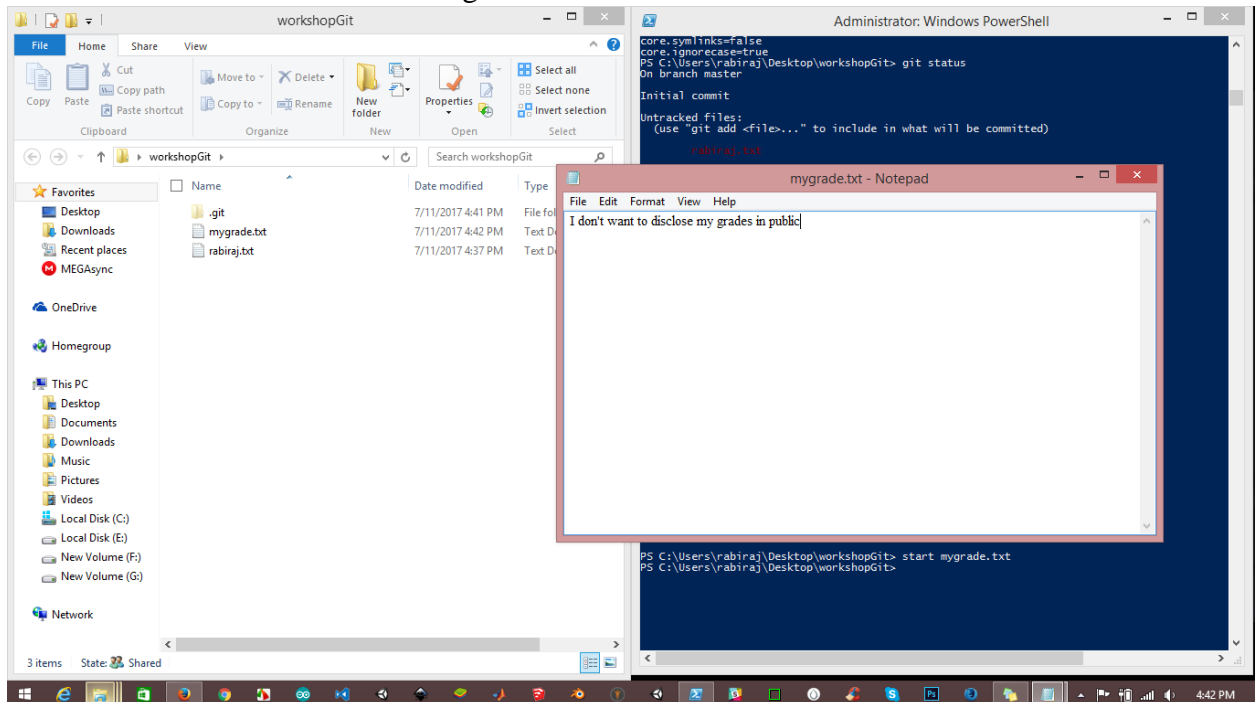
To view log of commit we use

*git log*

to view log of commit without user info we could type

*git log --oneline*

Till now we have committed two files rabiraj.txt and mygrades.txt in the master branch which is also a default branch after git is initialized.



```

PS C:\Users\rabiraj\Desktop\workshopGit> git add rabiraj.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "initial commit first file"
[master (root-commit) e6097fc] initial commit first file
1 file changed, 1 insertion(+)
create mode 100644 rabiraj.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git log
commit e6097fc5e510773cd31a2eb36ed3d8e897a86cbe
Author: Rabi Raj Khadka <er.rabirajkhadka@gmail.com>
Date: Tue Jul 11 16:41:23 2017 +0545

    initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> New-item mygrade.txt -type file

    Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   4:42 PM              0 mygrade.txt

PS C:\Users\rabiraj\Desktop\workshopGit> ls

    Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   4:42 PM              0 mygrade.txt
-a---             7/11/2017   4:37 PM             79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> start mygrade.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    mygrade.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   mygrade.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "grade status added"
[master 08a9445] grade status added
1 file changed, 1 insertion(+)
create mode 100644 mygrade.txt

```

```

[master 08a9445] grade status added
1 file changed, 1 insertion(+)
create mode 100644 mygrade.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git log
commit 08a9445b3689625d1c81ad98800040122f79425c
Author: Rabi Raj Khadka <er.rabirajkhadka@gmail.com>
Date: Tue Jul 11 16:43:32 2017 +0545

    grade status added

commit e6097fc5e510773cd31a2eb36ed3d8e897a86cbe
Author: Rabi Raj Khadka <er.rabirajkhadka@gmail.com>
Date: Tue Jul 11 16:41:23 2017 +0545

    initial commit first file

```

## 5. Branch and Merge

While working in real projects we use branch to work on our machine and changes necessary stuffs, test it and after it passes all test and after being ready to deliver to the client or public review we merge it to the master branch

Concept of branch also saves us from DETACHED HEAD which is the condition arise when we checkout the previous commit instead of branch and after some commit switching to some other branch makes the recent commit disappears as it does not belong to any branch

To make new branch we could use

`git branch <branch name>` or

`git checkout -b <branch name>`

The difference is that latter one will switch to new branch we just created i.e <branch name> but not in case of first one

```
initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> git checkout -b development
Switched to a new branch 'development'
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
* development
  master
PS C:\Users\rabiraj\Desktop\workshopGit> git checkout master
Switched to branch 'master'
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   4:42 PM             44 mygrade.txt
-a---             7/11/2017   4:37 PM             79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git checkout development
Switched to branch 'development'
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   4:42 PM             44 mygrade.txt
-a---             7/11/2017   4:37 PM             79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> start mygrade.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch development
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:   mygrade.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "lowest and highest grade added"
[development 18dbb0d] lowest and highest grade added
1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch development
nothing to commit, working tree clean
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
* development
  master
```

The changes in the branch is not globally reflected so changes are isolated from one branch to another which could also be verify by looking at the size of file in following figure at different branch.

```
nothing to commit, working tree clean
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
* development
  master
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---              7/11/2017   4:46 PM           199 mygrade.txt
-a---              7/11/2017   4:37 PM            79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git checkout master
Switched to branch 'master'
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---              7/11/2017   4:47 PM            44 mygrade.txt
-a---              7/11/2017   4:37 PM            79 rabiraj.txt

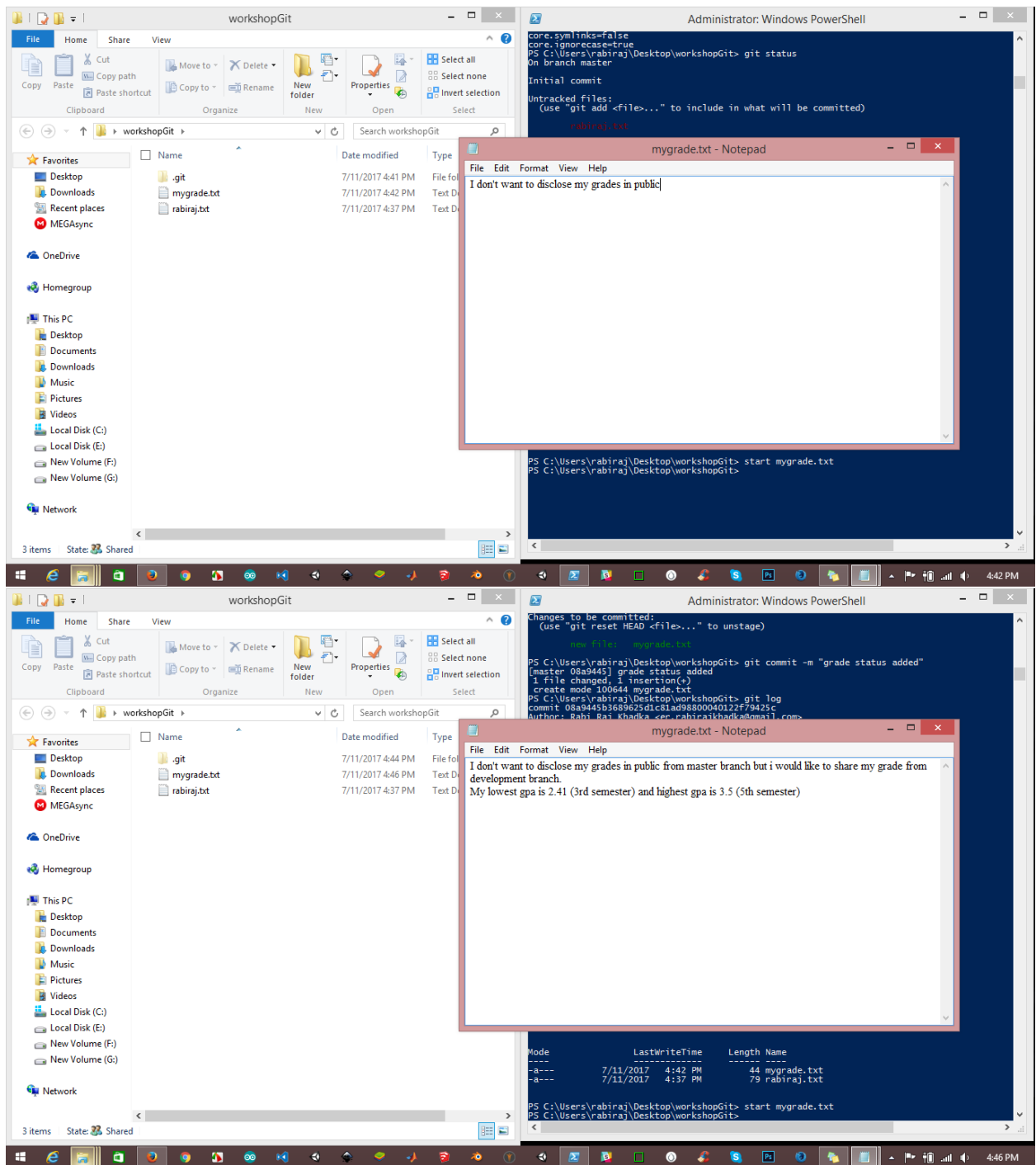
PS C:\Users\rabiraj\Desktop\workshopGit> start mygrade.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git merge development
Updating 08a9445..18dbb0d
Fast-forward
 mygrade.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---              7/11/2017   4:49 PM           199 mygrade.txt
-a---              7/11/2017   4:37 PM            79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git branch
  development
* master
PS C:\Users\rabiraj\Desktop\workshopGit> git branch -D development
Deleted branch development (was 18dbb0d).
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
* master
```





To merge the branch we use

*git merge <branch\_to\_be\_merged\_in\_current\_branch>*

The best practice is to delete the branch we no longer need and we could do so by using

*git branch -D <branch name to delete>* this is also shown in above figure

We could also reset the recent commit and go back to the previous commits using **reset**. Here I have created new file named myprojects.txt which is updated multiple times and two different commits have been committed. One with commit message “projects added” and another with “projects modified”

```
Select Administrator: Windows PowerShell

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

myprojects.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git tag v0.0.2 -a -m "second version with projects"
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "projects added"
[master 870309f] projects added
1 file changed, 8 insertions(+)
create mode 100644 myprojects.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\rabiraj\Desktop\workshopGit> git log --oneline
870309f projects added
18dbb0d lowest and highest grade added
08a9445 grade status added
e6097fc initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> start myprojects.txt
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   4:49 PM             199 mygrade.txt
-a---             7/11/2017   4:59 PM             315 myprojects.txt
-a---             7/11/2017   4:37 PM              79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "projets modified"
[master c8400b6] projets modified
1 file changed, 1 insertion(+), 3 deletions(-)
PS C:\Users\rabiraj\Desktop\workshopGit> git log --oneline
c8400b6 projets modified
870309f projects added
18dbb0d lowest and highest grade added
08a9445 grade status added
e6097fc initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> git tag
v0.0.1
v0.0.2
PS C:\Users\rabiraj\Desktop\workshopGit> git show v0.0.1
tag v0.0.1
Tagger: Rabi Raj Khadka <er.rabirajkhadka@gmail.com>
Date:   Tue Jul 11 16:50:35 2017 +0545

first version upto grade

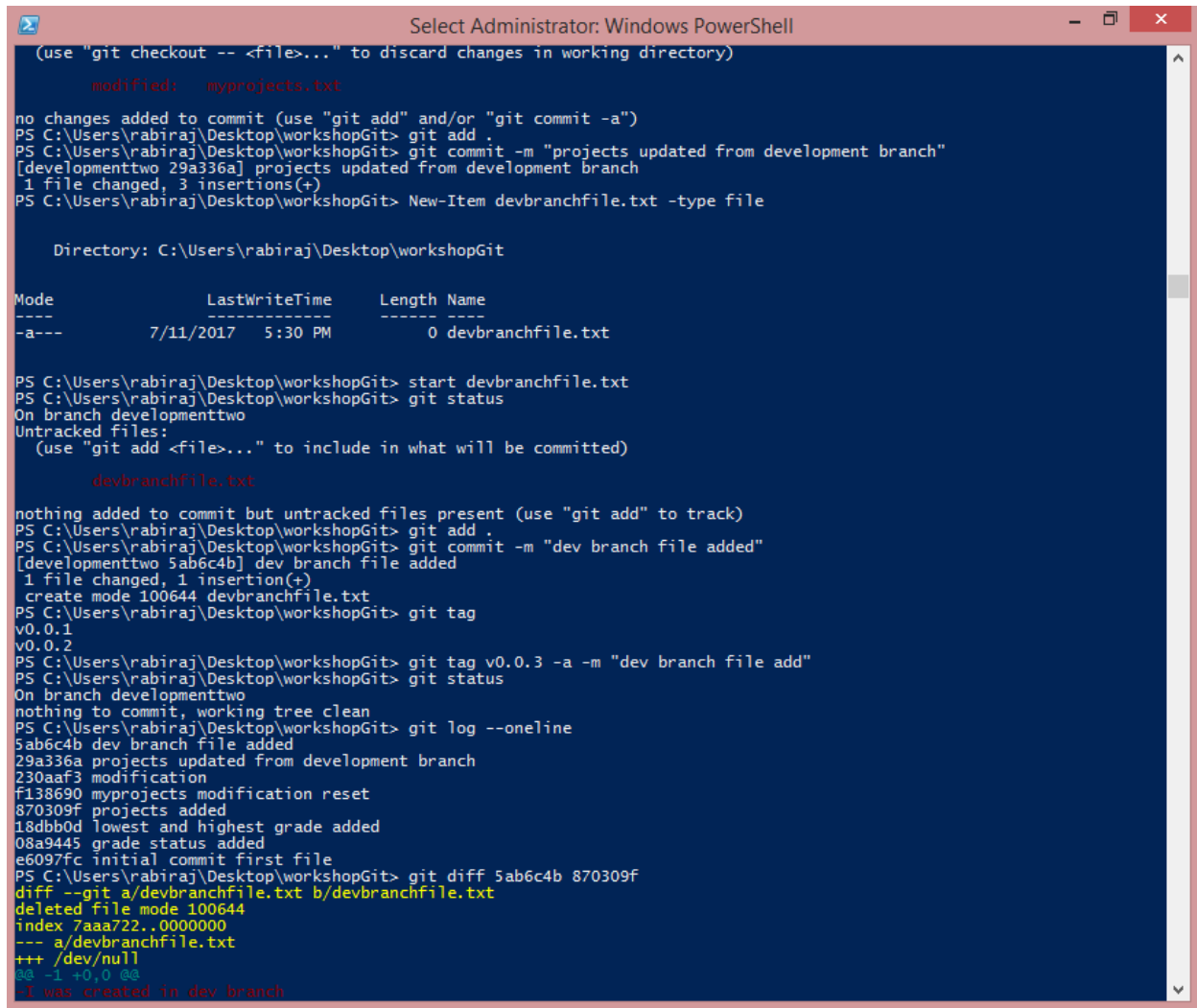
commit 18dbb0d2d6f2947ad482a645403503ec8dd918cd
Author: Rabi Raj Khadka <er.rabirajkhadka@gmail.com>
Date:   Tue Jul 11 16:46:58 2017 +0545

lowest and highest grade added
```

And then reset is used with hash of the commit as *git reset 870309f*

```
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
* master
PS C:\Users\rabiraj\Desktop\workshopGit> git log --oneline
c8400b6 projets modified
870309f projects added
18dbb0d lowest and highest grade added
08a9445 grade status added
e6097fc initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> git reset 870309f
Unstaged changes after reset:
M   myprojects.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git log --oneline
870309f projects added
18dbb0d lowest and highest grade added
08a9445 grade status added
e6097fc initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> start myprojects.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
```

To demonstrate more on how branch works we created new files and see the changes in branch developmenttwo and master.



```
(use "git checkout -- <file>..." to discard changes in working directory)

modified:   myprojects.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "projects updated from development branch"
[developmenttwo 29a336a] projects updated from development branch
1 file changed, 3 insertions(+)
PS C:\Users\rabiraj\Desktop\workshopGit> New-Item devbranchfile.txt -type file

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017   5:30 PM              0 devbranchfile.txt

PS C:\Users\rabiraj\Desktop\workshopGit> start devbranchfile.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch developmenttwo
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        devbranchfile.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "dev branch file added"
[developmenttwo 5ab6c4b] dev branch file added
1 file changed, 1 insertion(+)
create mode 100644 devbranchfile.txt
PS C:\Users\rabiraj\Desktop\workshopGit> git tag
v0.0.1
v0.0.2
PS C:\Users\rabiraj\Desktop\workshopGit> git tag v0.0.3 -a -m "dev branch file add"
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch developmenttwo
nothing to commit, working tree clean
PS C:\Users\rabiraj\Desktop\workshopGit> git log --oneline
5ab6c4b dev branch file added
29a336a projects updated from development branch
230aaf3 modification
f138690 myprojects modification reset
870309f projects added
18dbb0d lowest and highest grade added
08a9445 grade status added
e6097fc initial commit first file
PS C:\Users\rabiraj\Desktop\workshopGit> git diff 5ab6c4b 870309f
diff --git a/devbranchfile.txt b/devbranchfile.txt
deleted file mode 100644
index 7aaa722..0000000
--- a/devbranchfile.txt
+++ /dev/null
@@ -1,0,0 @@
-I was created in dev branch
```

To compare between two commits

*git diff <hashofcommitone> <hashofcommittwo>* as above, and to compare the difference between two branches we could use

*git diff <branch\_name\_tocompare>*

```
Select Administrator: Windows PowerShell

--- a/devbranchfile.txt
+++ /dev/null
@@ -1,0,0 @@
+I was created in dev branch
\ No newline at end of file
diff --git a/myprojects.txt b/myprojects.txt
index a75198a..88d7a13 100644
--- a/myprojects.txt
+++ b/myprojects.txt
@@ -2,7 +2,7 @@ my different projects are as below ordered by recent project first
1. Nepal Builder- Augmented Reality Game
2. Khwopa Canteen Food Order App- food ordering android app suitable for khwopa canteen
3. Escape Jump- android game where users have to run and collect the coins
+4. E-Library- ASP.NET web form application with lots of resources
+5. Data Logger- IoT project logging for the temperature of Server room of Khwop Engineering College
+6. Line Tracking Robot- black line following robot w/o programming, using sensors and motor drivers only
+4. E-Library- ASP.NET web form application with lots of resources(books,notes,assignments,labreports)
+5. Data Logger- IoT project logging an information (temperature) from MIS section
sensors and motordrivers.
and many more.....
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch developmenttwo
nothing to commit, working tree clean
PS C:\Users\rabiraj\Desktop\workshopGit> git diff master
diff --git a/devbranchfile.txt b/devbranchfile.txt
new file mode 100644
index 0000000..7aaa722
--- /dev/null
+++ b/devbranchfile.txt
@@ -0,0 +1 @@
+I was created in dev branch
\ No newline at end of file
diff --git a/myprojects.txt b/myprojects.txt
index ea329b1..a75198a 100644
--- a/myprojects.txt
+++ b/myprojects.txt
@@ -2,4 +2,7 @@ my different projects are as below ordered by recent project first
1. Nepal Builder- Augmented Reality Game
2. Khwopa Canteen Food Order App- food ordering android app suitable for khwopa canteen
3. Escape Jump- android game where users have to run and collect the coins
+4. E-Library- ASP.NET web form application with lots of resources
+5. Data Logger- IoT project logging for the temperature of Server room of Khwop Engineering College
+6. Line Tracking Robot- black line following robot w/o programming, using sensors and motor drivers only
and many more.....
PS C:\Users\rabiraj\Desktop\workshopGit> git checkout master
Switched to branch 'master'
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017  4:49 PM             199 mygrade.txt
-a---             7/11/2017  5:33 PM             298 myprojects.txt
-a---             7/11/2017  4:37 PM              79 rabiraj.txt
```

To host the github pages for free we then created index.html and added few codes and committed as below

```
PS C:\Users\rabiraj\Desktop\workshopGit> git branch
developmenttwo
* master
PS C:\Users\rabiraj\Desktop\workshopGit> New-Item Index.html -type file

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a---             7/11/2017  5:44 PM              0 Index.html

PS C:\Users\rabiraj\Desktop\workshopGit> Invoke-Item Index.html
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Index.html

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\rabiraj\Desktop\workshopGit> git add .
PS C:\Users\rabiraj\Desktop\workshopGit> git tag v1.0.0
PS C:\Users\rabiraj\Desktop\workshopGit> git commit -m "webpage created"
[master bde49c3] webpage created
1 file changed, 41 insertions(+)
create mode 100644 Index.html
PS C:\Users\rabiraj\Desktop\workshopGit> git status
```

## 6. Remote Repository

To work with remote repository we can use *git clone* and *git remote* commands. *git clone* will clone the remote repository to the local repository and establish the connection which we usually use at initial state before working on local. Today we will be using *git remote* to establish connection with the remote repository but before that let's create an account in Github and create one repository to push our codes in that repository.

The image is a composite of three screenshots from the GitHub website, illustrating the process of creating a new repository.

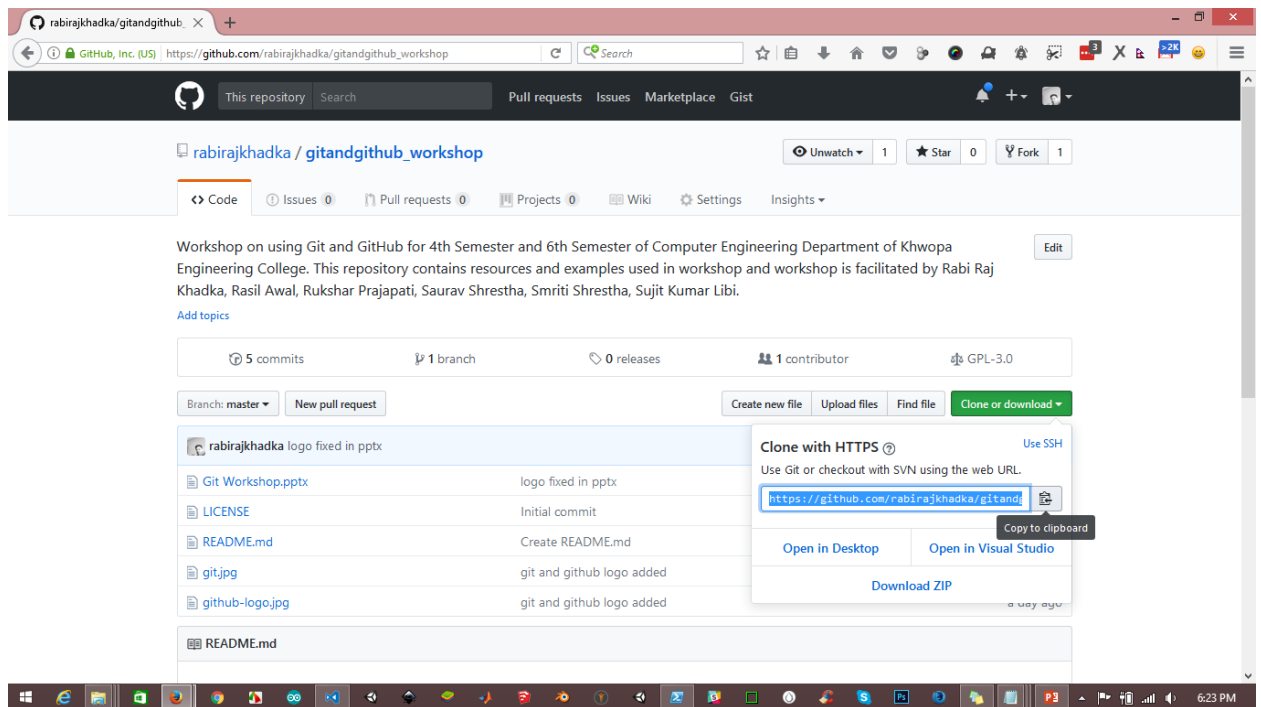
**Top Left Screenshot:** The GitHub sign-in page. It features the GitHub logo, the text "Sign in to GitHub", and a form with fields for "Username or email address" (containing "er.rabirajkhadka@gmail.com") and "Password". A "Sign in" button is at the bottom. Below the form, it says "New to GitHub? [Create an account.](#)".

**Top Right Screenshot:** The "Create a new repository" page. It shows the "Owner" as "rabirajkhadka" and the "Repository name" as "gitgithub\_workshop". The "Description (optional)" field contains "Git Workshop in Khwopa Engineering College". The "Public" radio button is selected. The "Initialize this repository with a README" checkbox is checked. A "Create repository" button is at the bottom.

**Bottom Screenshot:** The repository page for "rabirajkhadka / gitgithub\_workshop". It shows the repository name, a description, and statistics (1 commit, 1 branch, 0 releases, 1 contributor, GPL-3.0 license). A "Clone or download" button is highlighted, and a dropdown menu is open showing the "Clone with HTTPS" option and the URL "https://github.com/rabirajkhadka/gitgithub\_workshop".

As I have already created another repository for this purpose I will be using that repository instead of this





After copying the web url we use

*git remote add <remoteShortName> url*

before start pushing our code we need to check if there is any fast forward changes in remote or not so to synchronize the local and remote repository we use *git pull* commands and after syncing we use *git push*

*git pull* simply downloads the content of remote repository to your local repository

*git pull <remoteShortName> <branchName>*

e.g: *git pull origin master*

*git push <remoteShortName> <branchName>*

e.g: *git push origin master*

```
Select Administrator: Windows PowerShell

-a--- 7/11/2017 5:36 PM 572 myprojects.txt
-a--- 7/11/2017 4:37 PM 79 rabiraj.txt

PS C:\Users\rabiraj\Desktop\workshopGit> git pull origin master --allow-unrelated-histories
From https://github.com/rabirajkhadka/gitandgithub_workshop
* branch master -> FETCH_HEAD
Merge made by the 'recursive' strategy.
Git Workshop.pptx | Bin 0 -> 4242030 bytes
LICENSE | 674 ++++++++
README.md | 9 +
git.jpg | Bin 0 -> 64644 bytes
github-logo.jpg | Bin 0 -> 18954 bytes
5 files changed, 683 insertions(+)
create mode 100644 Git Workshop.pptx
create mode 100644 LICENSE
create mode 100644 README.md
create mode 100644 git.jpg
create mode 100644 github-logo.jpg
PS C:\Users\rabiraj\Desktop\workshopGit> ls

Directory: C:\Users\rabiraj\Desktop\workshopGit

Mode                LastWriteTime         Length Name
----                -
-a--- 7/11/2017 5:36 PM             27 devbranchfile.txt
-a--- 7/11/2017 6:30 PM        4242030 Git Workshop.pptx
-a--- 7/11/2017 6:30 PM         64644 git.jpg
-a--- 7/11/2017 6:30 PM         18954 github-logo.jpg
-a--- 7/11/2017 6:17 PM          1688 Index.html
-a--- 7/11/2017 6:30 PM         35815 LICENSE
-a--- 7/11/2017 4:49 PM           199 mygrade.txt
-a--- 7/11/2017 5:36 PM          572 myprojects.txt
-a--- 7/11/2017 4:37 PM           79 rabiraj.txt
-a--- 7/11/2017 6:30 PM          701 README.md

PS C:\Users\rabiraj\Desktop\workshopGit> git merge origin origin/master
merge: origin - not something we can merge
PS C:\Users\rabiraj\Desktop\workshopGit> git push origin master
Username for 'https://github.com': er.rabirajkhadka@gmail.com
Password for 'https://er.rabirajkhadka@gmail.com@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/rabirajkhadka/gitandgithub_workshop.git/'
PS C:\Users\rabiraj\Desktop\workshopGit> git push origin master
Username for 'https://github.com': rabirajkhadka
Password for 'https://rabirajkhadka@github.com':
Counting objects: 29, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (26/26), done.
Writing objects: 100% (29/29), 4.10 KiB | 0 bytes/s, done.
Total 29 (delta 9), reused 0 (delta 0)
remote: Resolving deltas: 100% (9/9), done.
To https://github.com/rabirajkhadka/gitandgithub_workshop.git
4f84dfd..38f4eb9 master -> master
PS C:\Users\rabiraj\Desktop\workshopGit> git tag
```

As we have used tag to identify the different releases i.e commit in local repository we also need to push the tags as by default git push command won't push the tags.

We use *git push - - tags* to update all tags in remote repository as shown below

```
Select Administrator: Windows PowerShell

To https://github.com/rabirajkhadka/gitandgithub_workshop.git
4f84dfd..38f4eb9 master -> master
PS C:\Users\rabiraj\Desktop\workshopGit> git tag
v0.0.1
v0.0.2
v0.0.3
v1.0.0
PS C:\Users\rabiraj\Desktop\workshopGit> git push --tags
Username for 'https://github.com': rabirajkhadka
Password for 'https://rabirajkhadka@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), 388 bytes | 0 bytes/s, done.
Writing objects: 100% (3/3), 388 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/rabirajkhadka/gitandgithub_workshop.git
* [new tag] v0.0.1 -> v0.0.1
* [new tag] v0.0.2 -> v0.0.2
* [new tag] v0.0.3 -> v0.0.3
* [new tag] v1.0.0 -> v1.0.0
PS C:\Users\rabiraj\Desktop\workshopGit> git status
On branch master
nothing to commit, working tree clean
```

## 7. Github Pages

We have updated the index.html file in remote repository so now we are ready to host the github page and it is super simple all we need to do is create new branch *gh-pages* and it is published in [username.github.io/repositoryname](https://username.github.io/repositoryname)

Go to the repository page and check if there is index.html is present or not if yes click on branch and type gh-pages and create new branch. As soon as you created this branch your page is setup.

rabirajkhadka / gitandgithub\_workshop

Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Workshop on using Git and GitHub for 4th Semester and 6th Semester of Computer Engineering Department of Khwopa Engineering College. This repository contains resources and examples used in workshop and workshop is facilitated by Rabi Raj Khadka, Rasil Awal, Rukshar Prajapati, Saurav Shrestha, Smriti Shrestha, Sujit Kumar Libi.

Add topics

15 commits 1 branch 4 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

rabirajkhadka Merge branch 'master' of https://github.com/rabirajkhadka/gitandgithub... Latest commit 38f4eb9 7 minutes ago

Git Workshop.pptx	logo fixed in pptx	a day ago
Index.html	webpage created	19 minutes ago
LICENSE	Initial commit	a day ago
README.md	Create README.md	a day ago
devbranchfile.txt	dev branch file added	an hour ago
git.jpg	git and github logo added	a day ago
github-logo.jpg	git and github logo added	a day ago

rabirajkhadka / gitandgithub\_workshop

Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Workshop on using Git and GitHub for 4th Semester and 6th Semester of Computer Engineering Department of Khwopa Engineering College. This repository contains resources and examples used in workshop and workshop is facilitated by Rabi Raj Khadka, Rasil Awal, Rukshar Prajapati, Saurav Shrestha, Smriti Shrestha, Sujit Kumar Libi.

Add topics

15 commits 1 branch 4 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

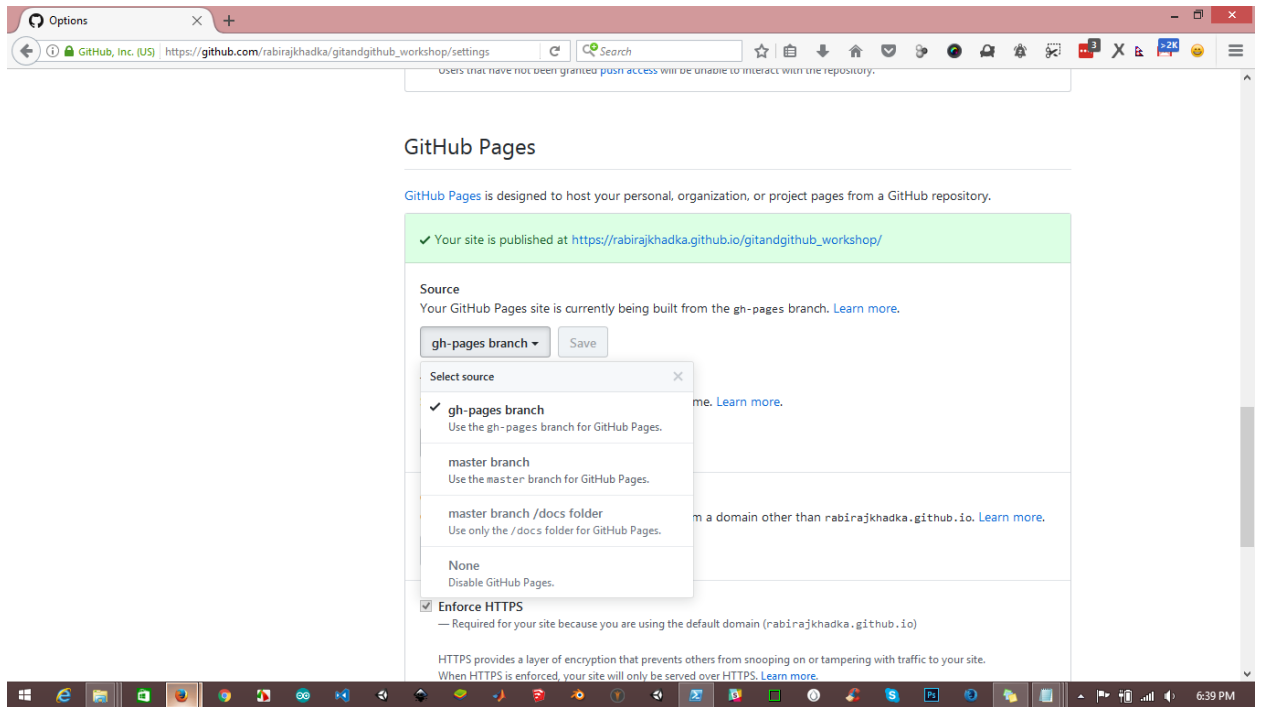
Switch branches/tags

gh-pages

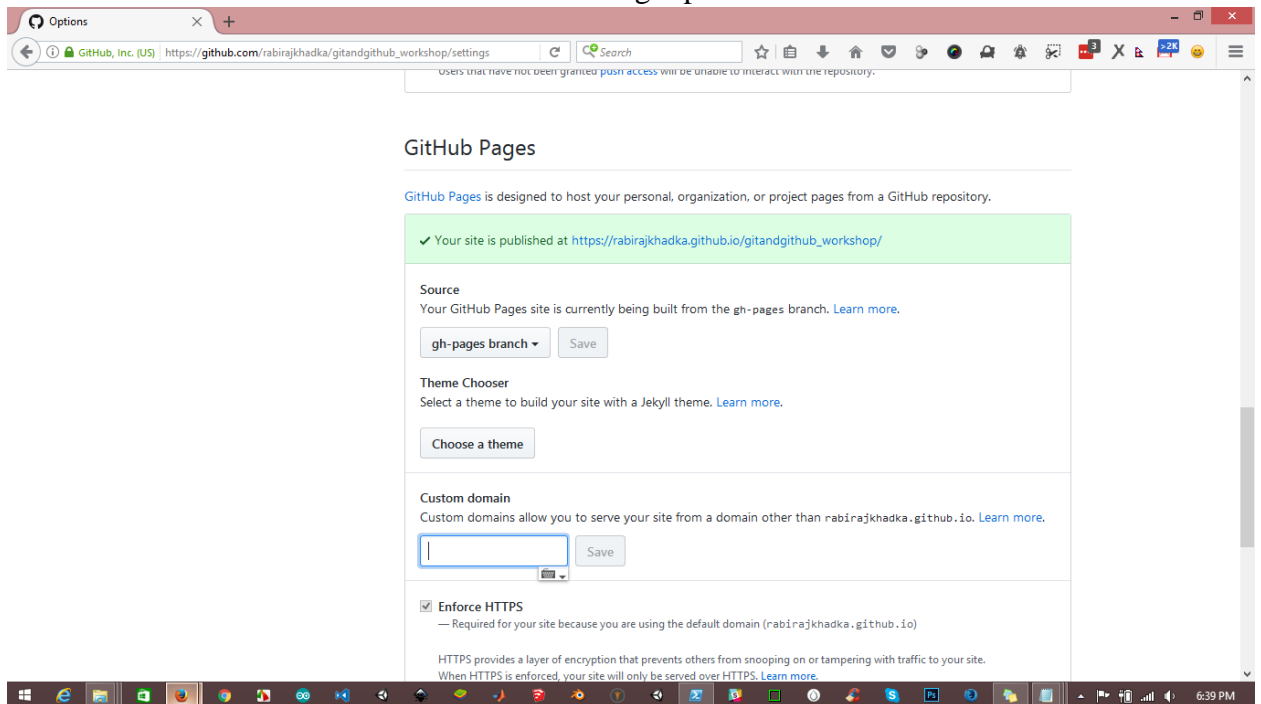
Create branch: gh-pages from 'master'

rabirajkhadka Merge branch 'master' of https://github.com/rabirajkhadka/gitandgithub... Latest commit 38f4eb9 7 minutes ago

Git Workshop.pptx	logo fixed in pptx	a day ago
Index.html	webpage created	19 minutes ago
LICENSE	Initial commit	a day ago
README.md	Create README.md	a day ago
devbranchfile.txt	dev branch file added	an hour ago
git.jpg	git and github logo added	a day ago
github-logo.jpg	git and github logo added	a day ago



To host custom domain we can add it in following input box.



## 8. Summarizing

- We first installed git in our system and verified its installation by checking version as `git --version` then initialized the git repository in folder at Desktop



- We created new files and added it to local repository using `git add` and `git commit` after configuring our `user.name` and `user.email` with `git config --global user.name "you name"`
- We used `git tag`, `git status`, `git log` commands to see the status of repository and commits on local repo.
- We created different branch and edited same file and checked if changes are reflected across branches or not
  - `git branch`
  - `git checkout -b branchName`
  - `git checkout branch`
  - `git merge anotherBranch`
  - `git branch -D branchName`
- We used `git diff` to compare between different branch and commits
  - `git diff commit1 commit2`
  - `git diff anotherBranch`
- We published our code to github repository if we use `git clone` then `remoteName` will be origin by default and we get flexibility to change it while using `git remote add`.
  - `git remote add remoteName remoteURL`
  - `git pull remoteName branchName`
  - `git push remoteName branchName`
- We then hosted our static web page using `gh-pages` branch in github

## 9. More Resources

- <https://git-scm.com/doc> which have good documentatin to use git
- <https://vimeo.com/41027679>
- <https://vimeo.com/41381741>
- <https://vimeo.com/41493906>
- <https://vimeo.com/41516942>
- <https://git-scm.com/downloads/guis> if you like to use gui versions (mine personal recommendation gitKraken and SourceTree)
- <https://www.atlassian.com/git/tutorials>
- <https://try.github.io/levels/1/challenges/1> practice your basic skill here
- <http://learngitbranching.js.org/> learn about branch in detail
- <https://git-scm.com/book/en/v2> very nice book about git also available in this repo.
- <https://education.github.com/pack> claim your student pack here

- <http://think-like-a-git.net/>
- [http://ericsink.com/vcbe/vcbe\\_a4\\_lo.pdf](http://ericsink.com/vcbe/vcbe_a4_lo.pdf) version controlling with centralized and distributed VCS (subversion, git and mercurial)
- <http://ftp.newartisans.com/pub/git.from.bottom.up.pdf>

As someone has asked us about memory we are allowed in github, i want to repeat answer again that github won't restrict in number of repositories but suggest us to keep repository under **1GB** each and warns us to push the file greater than **50MB** and limit the file size to **100MB** in addition if we use ftp to upload file in github the file size is restricted to **25MB**

Few commands we haven't gone thorough in detail are

reflog, stash, cherry-pick, instaweb, archive, gc ,prune, fsck, and many more