

GIT AND GITHUB

Tutorial 4

What is Git and Github?

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Github is cloud based hosting Service which allows you to manage git repositories.

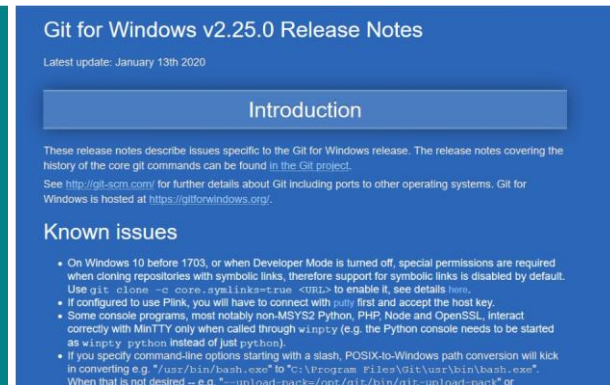
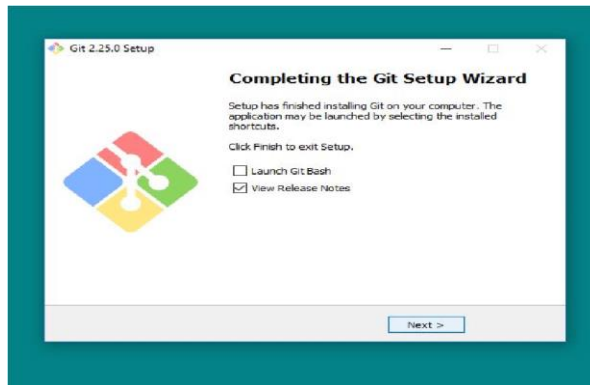
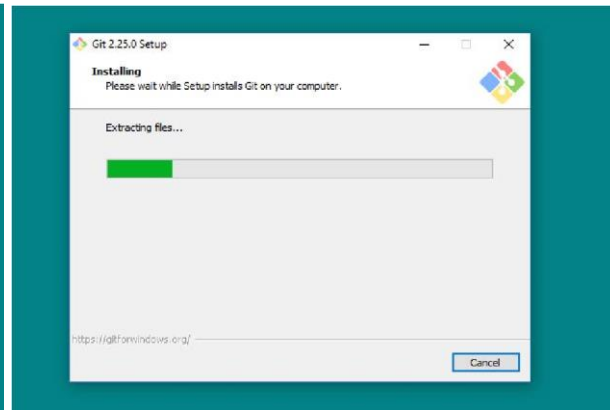
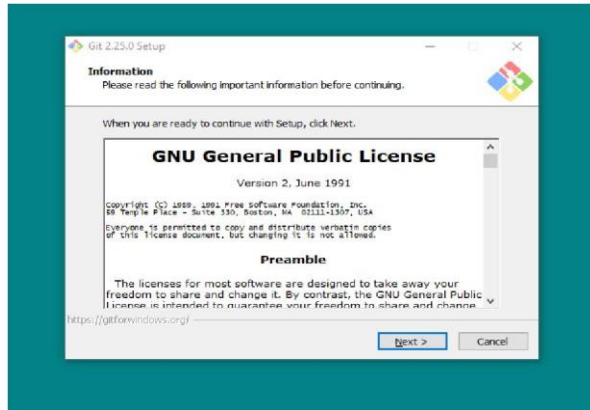
How to install Git?

Mac : <http://git-scm.com/download/mac>

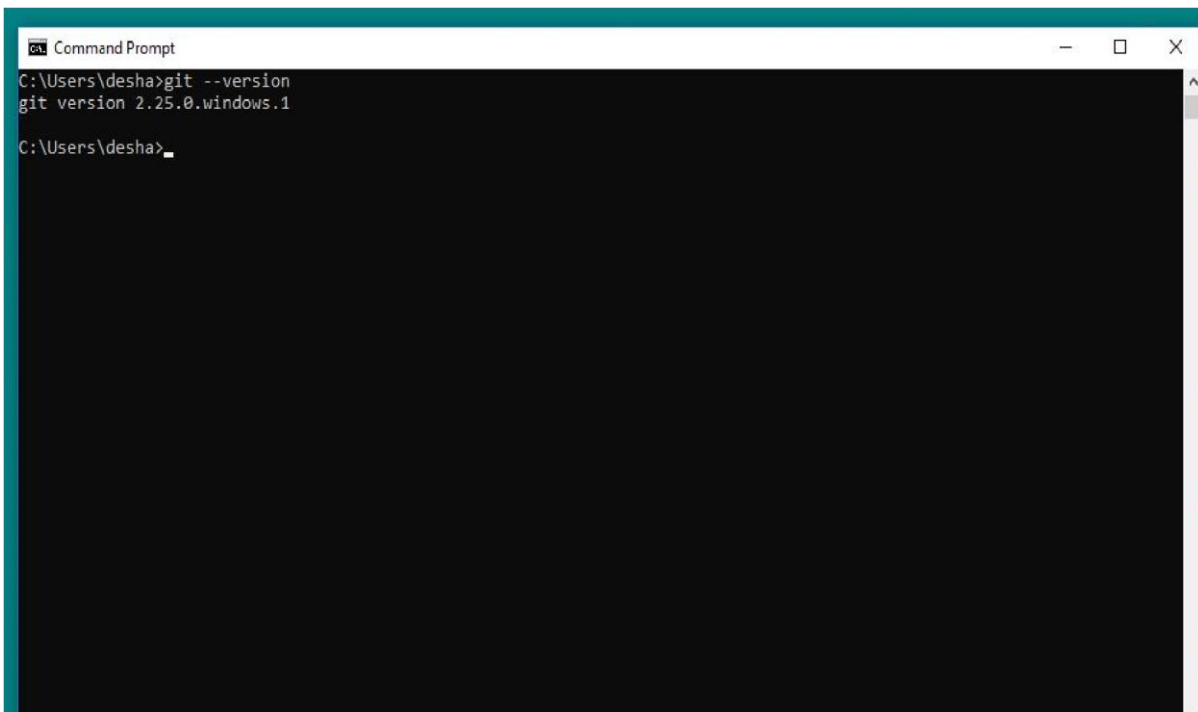
Windows: <http://git-scm.com/download/win>

The screenshot shows the Git website with the tagline "--distributed-is-the-new-centralized". The left sidebar contains links for "About", "Documentation" (with sub-links: Reference, Book, Videos, External Links), "Downloads", and "Community". Below these are links for translations: "This book is available in English.", "Full translation available in", and a list of languages including Bulgarian, Spanish, French, Greek, Japanese, Korean, Dutch, Russian, and Slovenian. The main content area is titled "1.5 Getting Started - Installing Git" and includes a "Note" about Git version 2.8.0 and instructions for installing on Linux using either dnf or apt.

The screenshot shows the Git website with the tagline "--distributed-even-if-your-workflow-isnt". The left sidebar contains links for "About", "Documentation", "Downloads" (with sub-links: GUI Clients, Logos), and "Community". Below these is a link to the "Pro Git" book. The main content area is titled "Downloading Git" and includes a section "Your download is starting..." with information about the latest 64-bit version of Git for Windows. It also provides links for "Other Git for Windows downloads" including 32-bit and 64-bit setups, and portable versions. A note at the bottom mentions the current source code release is version 2.25.0.

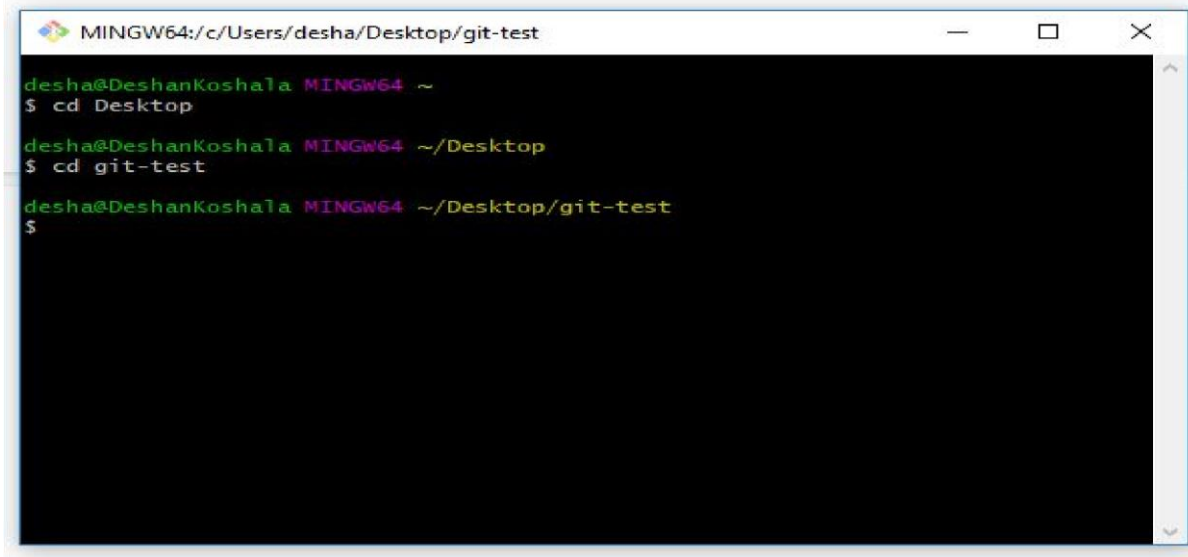


Use `git --version` to find your version.



How To use git?

1. Create a folder "git-test" in Desktop.
2. Open Git Bash.
3. Change your working directory to git-test.



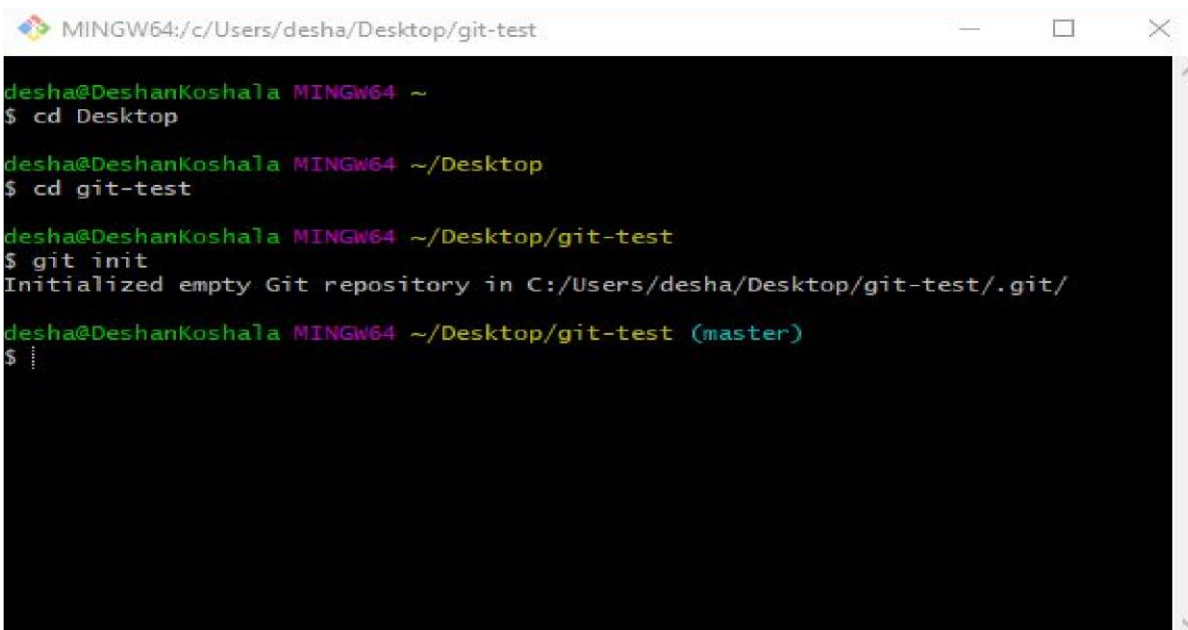
```
MINGW64:/c/Users/desha/Desktop/git-test

desha@DeshanKoshala MINGW64 ~
$ cd Desktop

desha@DeshanKoshala MINGW64 ~/Desktop
$ cd git-test

desha@DeshanKoshala MINGW64 ~/Desktop/git-test
$
```

4. Use command **git init** command to creates a new **Git** repository



```
MINGW64:/c/Users/desha/Desktop/git-test

desha@DeshanKoshala MINGW64 ~
$ cd Desktop

desha@DeshanKoshala MINGW64 ~/Desktop
$ cd git-test

desha@DeshanKoshala MINGW64 ~/Desktop/git-test
$ git init
Initialized empty Git repository in C:/Users/desha/Desktop/git-test/.git/

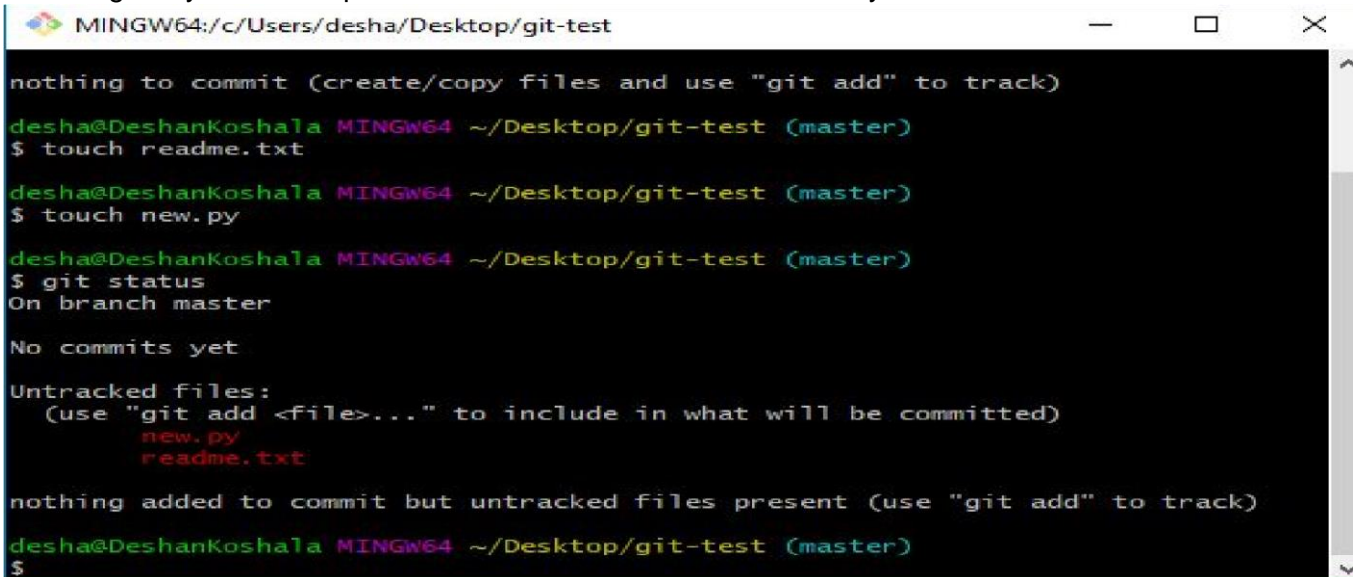
desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$
```

5. Create 2 files in the folder git-test

1. Readme.txt
2. new.py

6. Check for **git status**

It gives you a description on commits and files in the directory.

A terminal window titled 'MINGW64:/c/Users/desha/Desktop/git-test' showing the output of the 'git status' command. The output indicates that there are no commits yet and that there are untracked files, 'new.py' and 'readme.txt', which need to be added to the commit.

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

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ touch readme.txt

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ touch new.py

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git status
On branch master

No commits yet

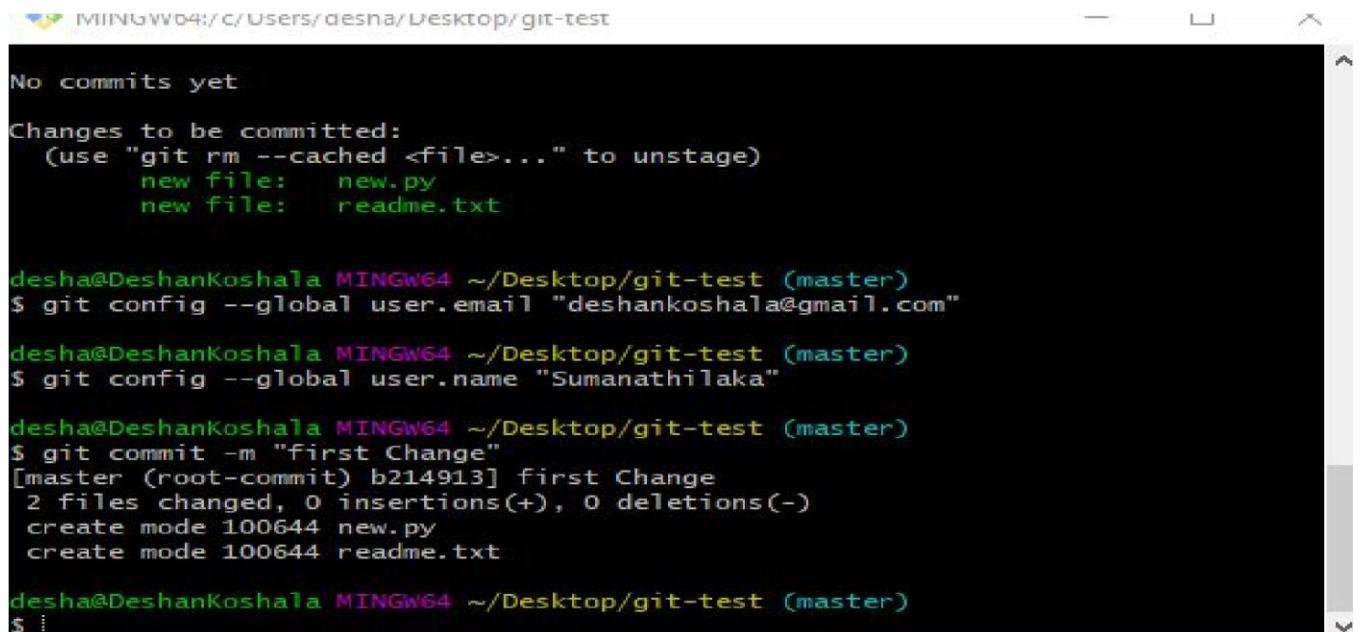
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        new.py
        readme.txt

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

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$
```

7. The “**git add**” command adds a change in the working directory to the staging area. It tells **Git** that you want to include updates to a particular file in the next commit.

8. Use Command **git commit -m “first change”** to save your changes to the local repository. If you are performing the above task for the first time you should add your username and email details.

A terminal window titled 'MINGW64:/c/Users/desha/Desktop/git-test' showing the output of 'git status' and the execution of 'git config' and 'git commit' commands. The output shows that the files 'new.py' and 'readme.txt' are staged for commit. The 'git config' command is used to set the global user email to 'deshankoshala@gmail.com' and the global user name to 'Sumanathilaka'. Finally, the 'git commit -m "first Change"' command is executed, creating a new commit with the message 'first Change' and including the two files.

```
No commits yet

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

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git config --global user.email "deshankoshala@gmail.com"

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git config --global user.name "Sumanathilaka"

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git commit -m "first Change"
[master (root-commit) b214913] first Change
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.py
 create mode 100644 readme.txt

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$
```

9. Do changes to your files `readme.txt` and `new.py`. Check for status. Use **git status**. Add files to the staging area and commit with the name `Second Change`.

```
MINGW64:/c:/Users/desha/Desktop/git-test
desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   readme.txt

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:   new.py
    modified:   readme.txt

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git add .

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git commit -m "Second Change"
[master 80f17bc] Second Change
 2 files changed, 3 insertions(+)

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ |
```

10. Use following commands to delete a file. **git rm --cached <filename>**

11. **Branch** in git. - In **Git**, branches are a part of your everyday development process. **Git** branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new **branch** to encapsulate your changes. Use command **git branch <Branch Name>**

```
desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git branch login

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (master)
$ git checkout login
Switched to branch 'login'

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (login)
$ git status
On branch login
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   readme.txt

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

desha@DeshanKoshala MINGW64 ~/Desktop/git-test (login)
$ git add .

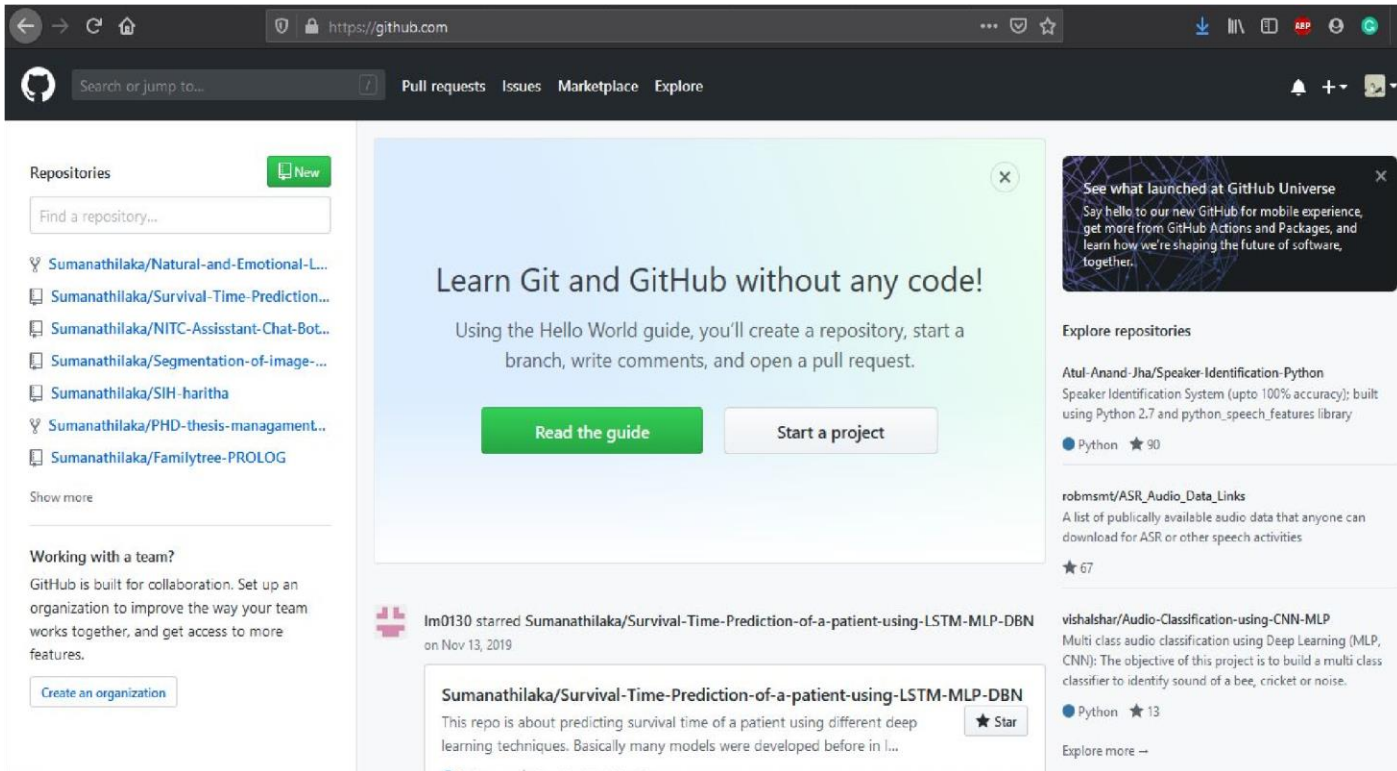
desha@DeshanKoshala MINGW64 ~/Desktop/git-test (login)
$ git commit -m "third change"
[login 01ffd5d] third change
 1 file changed, 1 insertion(+), 2 deletions(-)
```

Use Command **git checkout <new Branch>** to switch the branch. Create a new file and also change the contents in the available files and do add and commit to the files. Now change the branch and identify the difference if each branch.

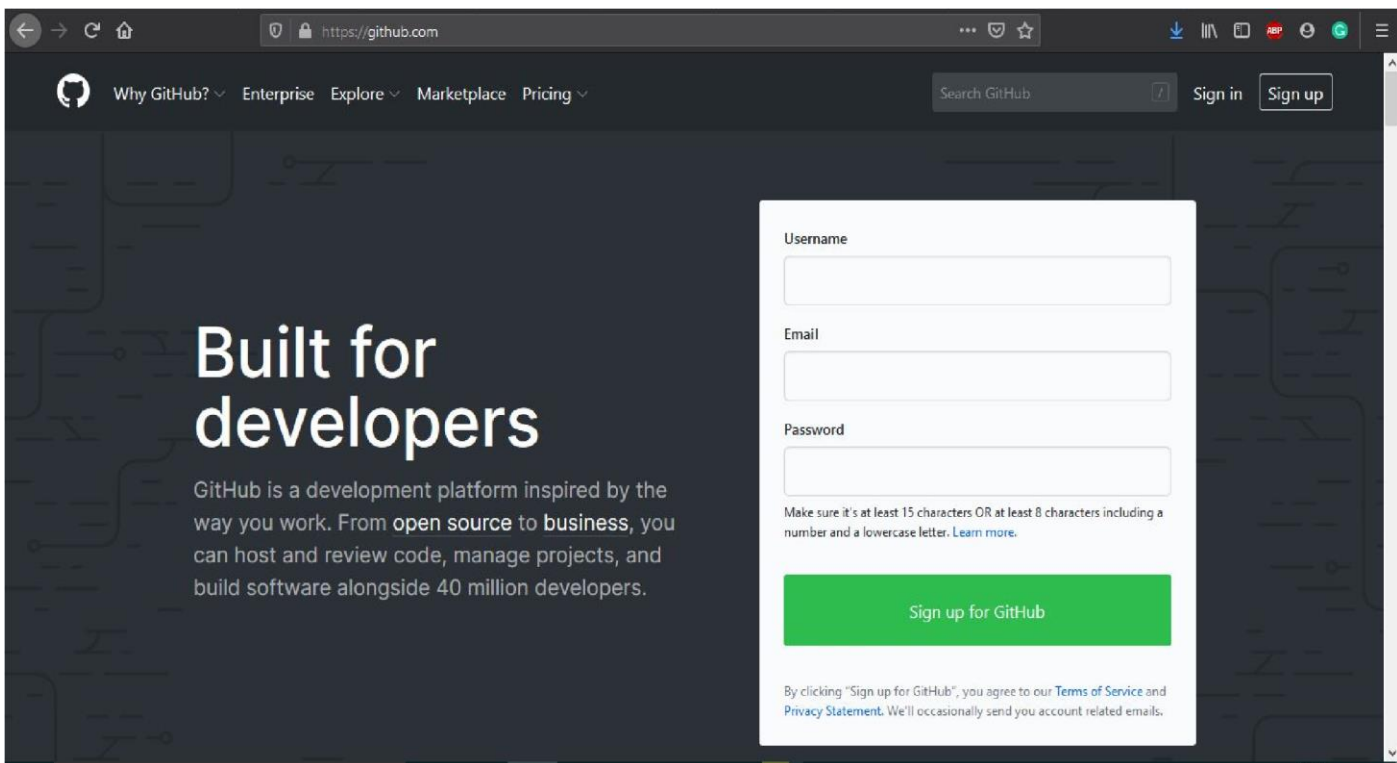
11. Find the Usage of command merge.

Git merge <>

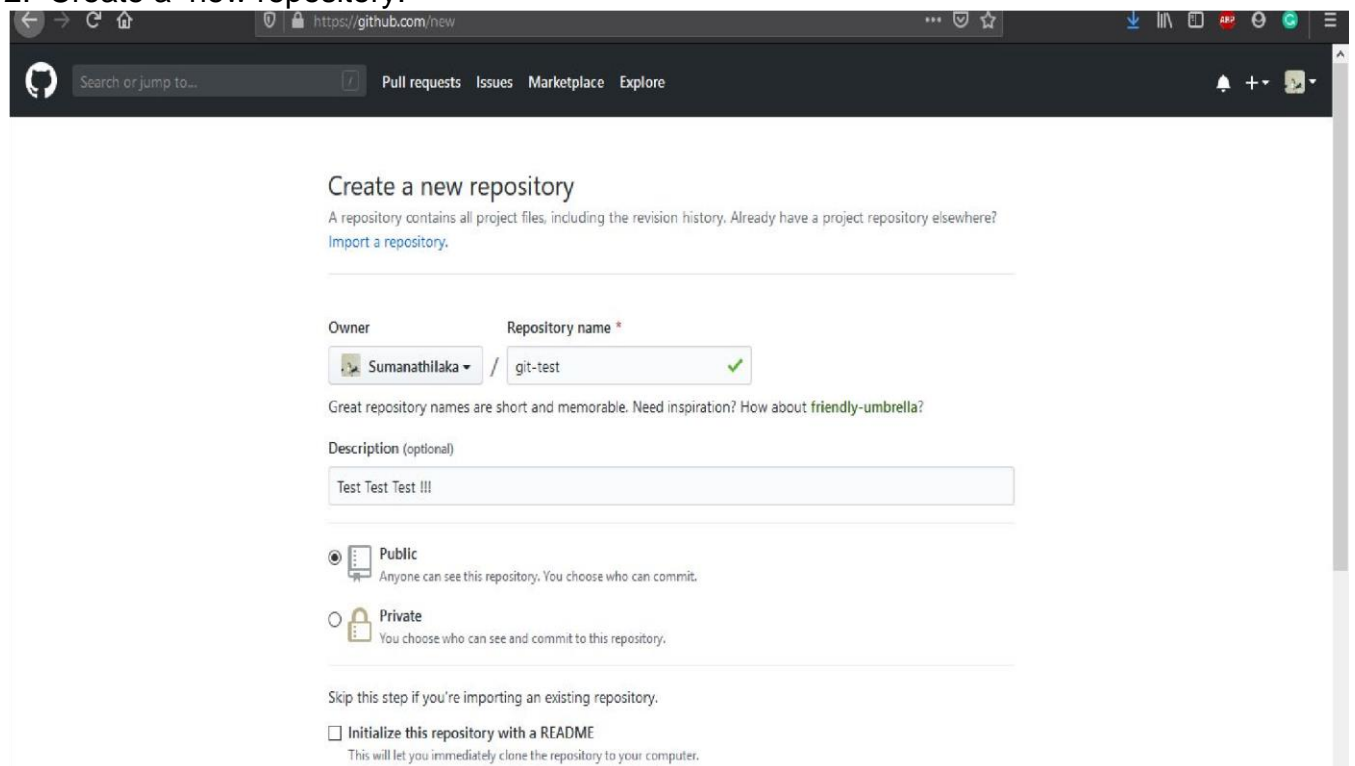
GIT HUB



1. Create an account in Github - <https://github.com/>



2. Create a new repository.

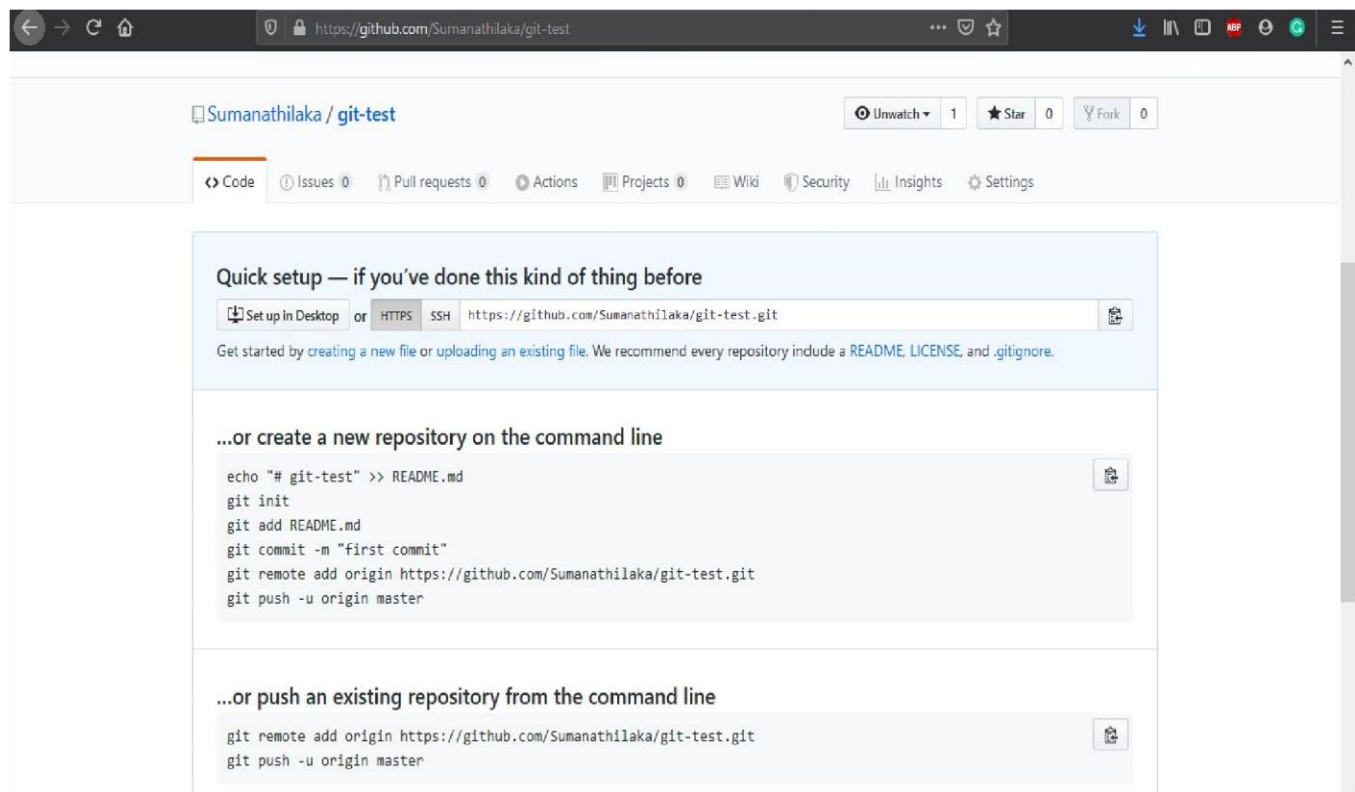


The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'https://github.com/new'. The page has a dark header with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The main content area is titled 'Create a new repository' and includes a sub-header: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

The form contains the following fields and options:

- Owner:** A dropdown menu showing 'Sumanathilaka'.
- Repository name:** A text input field containing 'git-test' with a green checkmark icon to its right.
- Description (optional):** A text input field containing 'Test Test Test !!!'.
- Visibility:** Two radio buttons. The 'Public' option is selected, with the text 'Anyone can see this repository. You choose who can commit.' below it. The 'Private' option is unselected, with the text 'You choose who can see and commit to this repository.' below it.
- Initialize this repository with a README:** An unchecked checkbox with the text 'This will let you immediately clone the repository to your computer.'

Below the form, there is a link: 'Skip this step if you're importing an existing repository.'



The screenshot shows the GitHub repository page for 'Sumanathilaka / git-test'. The browser address bar shows 'https://github.com/Sumanathilaka/git-test'. The page has a light gray header with the repository name and links for 'Unwatch', 'Star' (1), and 'Fork' (0). Below the header is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'.

The main content area is titled 'Quick setup — if you've done this kind of thing before' and includes a text input field for the repository URL: 'https://github.com/Sumanathilaka/git-test.git'. Below this is a link: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).' Below this is a section titled '...or create a new repository on the command line' with a code block containing the following commands:

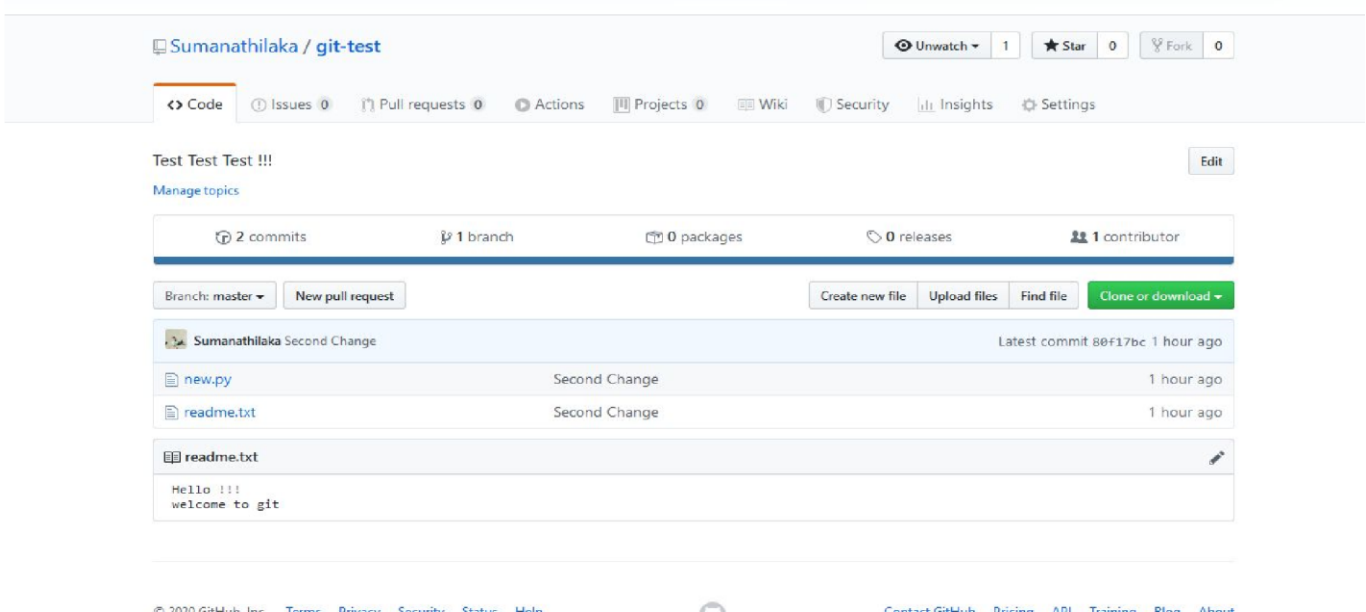
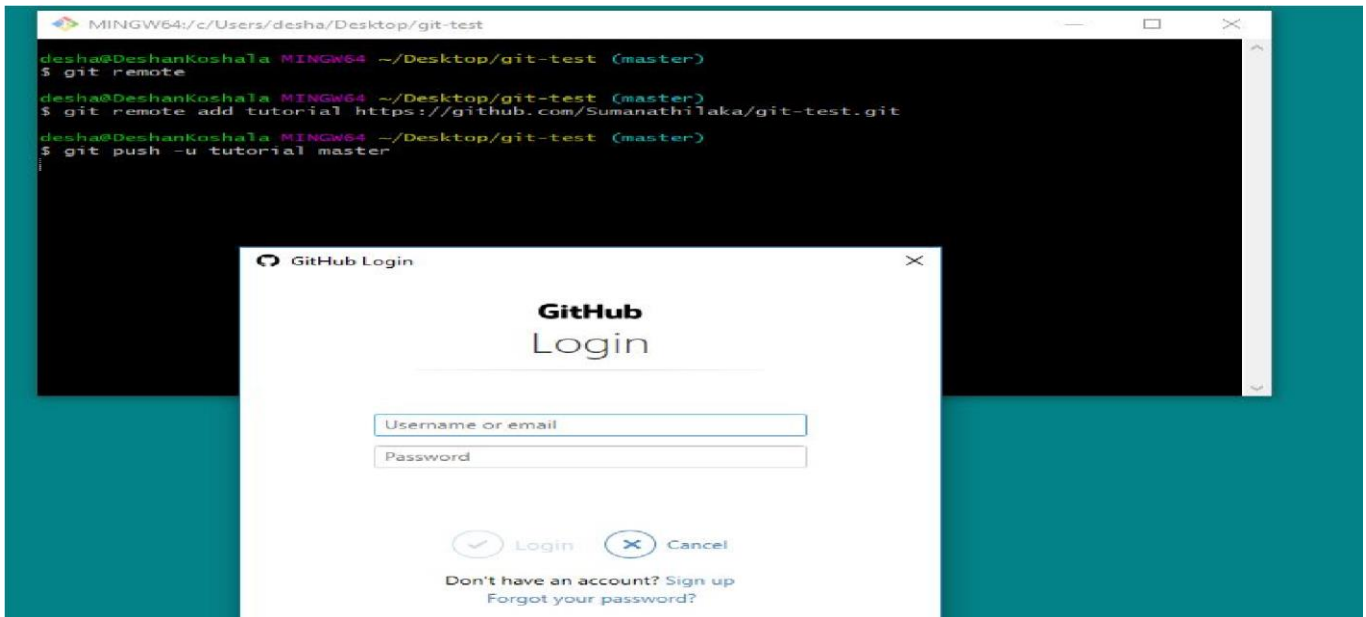
```
echo "# git-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/Sumanathilaka/git-test.git
git push -u origin master
```

Below this is a section titled '...or push an existing repository from the command line' with a code block containing the following commands:

```
git remote add origin https://github.com/Sumanathilaka/git-test.git
git push -u origin master
```


3. Use the following Commands.

```
git remote git remote add  
<name> <gitlink> git push -u  
<name> master
```



Add both master and login branches to github and check.

