

# Git & Github

## Download and install git:

Download from : [git-scm.com](http://git-scm.com)

Gitbash is a terminal based on Unix OS.

### General Concept :

- \* **Working area --> Staging area --> Repository**
- \* **Push** --> Act of sending file to github remote server from local PC
- \* **Pull** --> Act of receiving changes from remote to local PC
- \* **Remote** --> Duplicate instance of repository on remote server
- \* **Clone** --> Act of downloading the repository to local PC

### Create github Account:

- Open [github.com](https://github.com)
- Create account
- Create New repository by clicking + icon with any valid repository name

### Open git from terminal:

- Command to check git version : git version

### Git configuration username and email:

- git config --global user.name "raju-shrestha"
- git config --global user.email "[rajushrestha2007@gmail.com](mailto:rajushrestha2007@gmail.com)"

### **Check the username and email verification:**

- `git config --global -list`

This command will display username and email which is configured.

### **Directory/location:**

- Check the Current directory : `pwd`
- Change directory : `cd directoryname`
- Change the directory to 1 step back: `cd ..`

### **Create Directory/Folder:**

- Create directory on PC (Let it be on Desktop).
- Command to create directory : `mkdir directoryname` (eg. Github)
- Put files in that directory which we are going to push.
- Change directory : `cd directoryname`

Command to initialize git and check status:

- `git init`
- `git status`

This command will show you a list of all untracked files and directories in your repository.

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git & Github.docx
    ~$t & Github.docx

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

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

Untracked files in Git are files present in your working directory that are not currently under Git's version control. This means Git does not know about their existence, and they are not part of any commit history.

### **Adding Untracked Files to Git:**

- `git add <filename>`

*To add all untracked files in the current directory*

- `git add . OR git add -A`

This moves the specified files from the untracked state to the staged state, preparing them for the next commit.

**Check the status of file after git add command:** `git status`

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  Git & Github.docx
  new file:  ~$t & Github.docx

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

**Git commit:** git commit -m "Commit Message"

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git commit -m "initial commit"
[master (root-commit) 6727140] initial commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Git & Github.docx
 create mode 100644 ~$t & Github.docx

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

**Check git status of current working direcory:** git status

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git status
On branch master
nothing to commit, working tree clean

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

**Check committed files in history:** git log

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git log
commit 67271406555da4e52acadd39699a7b72687bb622 (HEAD -> master)
Author: raju-shrestha <rajushrestha2007@gmail.com>
Date:   Wed Oct 29 15:30:43 2025 +0545

  initial commit

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

**To clear git command from terminal:** clear

**Remove Unnecessary file:**

- git rm ‘filename’

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git rm '~$t & Github.docx'
rm '~$t & Github.docx'

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ ls
'Git & Github.docx'
```

**To make separate branch:**

- git branch branchname

**Delete Branch:**

- git branch -d branchname

**List all the branch in the current repository:** git branch

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git branch
* master

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ ...
```

**Switching/Change between the branch:** git checkout branchname

**Merge two branch:**

- First go to master branch: git checkout master
- Then **git merge branchname** which you want to merge.

## Create New repository in git server after creating github account

- Click on + icon to create new repository.

The screenshot shows the GitHub 'Create a new repository' interface. At the top, there's a header with navigation icons and a search bar. Below it, a sidebar menu is open, showing options like 'New issue', 'New repository' (which is selected), 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. The main form has two sections: 'General' and 'Configuration'. In the 'General' section, 'Owner' is set to 'raju-shrestha' and 'Repository name' is empty. A note says 'Great repository names are short and memorable. How about [literate-waddle](#)?'. In the 'Description' field, it says '0 / 350 characters'. In the 'Configuration' section, 'Choose visibility' is set to 'Public'. There are options to 'Add README' (which is off) and 'Add .gitignore' (which is set to 'No .gitignore').

## Connect local git repository to remote repository:

- [git remote add origin https://github.com/raju-shrestha/Git-Github.git](#)

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git remote add origin https://github.com/raju-shrestha/Git-Github.git

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git remote
origin

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$
```

**Git push origin to master branch:** git push -u origin master

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (1/1), done.
Writing objects: 100% (2/2), 239 bytes | 239.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/raju-shrestha/Git-Github.git
  6727140..ec10a15 master -> master
branch 'master' set up to track 'origin/master'.
```

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ |
```

Now clone the pushed repository

**Git Clone:**

- Copy/download the repository from github to your local PC called git clone.
- Change the directory where you want to clone the repository: cd directoryname
- [git clone https://github.com/raju-shrestha/Git-Github.git](https://github.com/raju-shrestha/Git-Github.git)

```
User@DESKTOP-SVKOMKA MINGW64 ~
$ pwd
/c/Users/User

User@DESKTOP-SVKOMKA MINGW64 ~
$ cd Desktop

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop
$ git clone https://github.com/raju-shrestha/Git-Github.git
Cloning into 'Git-Github'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 12.55 KiB | 291.00 KiB/s, done.

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop
$ |
```

**Make some changes in this file and check the status: git status**

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (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 restore <file>..." to discard changes in working directory)
    modified:   Git & Github.docx

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ~$t & Github.docx

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

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ |
```

**The modified file is not commited so first add it to stagging area.**

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ git add .

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Git & Github.docx
    new file:   ~$t & Github.docx

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ |
```

**Commit the changes:** git commit -m “Commit Message”

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ git commit -m "Fourth commit"
[master 36d363f] Fourth commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ~\$t & Github.docx

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ |
```

**Push the committed file:** git push -u origin master

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 16.49 KiB | 2.75 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/raju-shrestha/Git-Github.git
  54019d3..36d363f  master -> master
branch 'master' set up to track 'origin/master'.

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Git-Github (master)
$ |
```

Finally, the changed file is push to github remote repository.

## Git Pull:

The git pull command in Git is used to fetch and download content from a remote repository and immediately update the local repository to match that content.

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 289 bytes | 10.00 KiB/s, done.
From https://github.com/raju-shrestha/Git-Github
  440fb24..b6c0ea2 master      -> origin/master
Updating 440fb24..b6c0ea2
Fast-forward
 Update.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Update.txt
```

## Add Collaborator:

A Git collaborator is an individual granted access to contribute to a Git repository, typically hosted on platforms like GitHub. This access allows them to make changes, push code, and actively participate in the development of a project.

For adding collaborator: Go to settings

The screenshot shows the GitHub repository settings page for 'Git-Github'. The left sidebar is titled 'General' and contains sections for 'Access' (with 'Collaborators' selected), 'Code and automation', and 'Pages'. The main area is titled 'Collaborators and teams' and shows a 'Public repository' status with a note that 0 collaborators have access. A 'Manage visibility' button is present. Below this is a 'Direct access' section with a note that only the repository owner can contribute. A 'Manage access' section shows a user icon and the message 'You haven't invited any collaborators yet'.

Go to collaborators then add people.

And give correct username or email address of collaborators who you want to make.

Accept the request from collaborator site.

Then clone the repository to your local PC.

Make some changes to file and then add, commit and push to remote repository.

```
User@DESKTOP-SVKOMKA MINGW64 ~/Downloads/Git-Github (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 restore <file>..." to discard changes in working directory)
    modified:   Update.txt

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

User@DESKTOP-SVKOMKA MINGW64 ~/Downloads/Git-Github (master)
$ git add .

User@DESKTOP-SVKOMKA MINGW64 ~/Downloads/Git-Github (master)
$ git commit -m "This is from collaborators"
[master 07fd994] This is from collaborators
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-SVKOMKA MINGW64 ~/Downloads/Git-Github (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/raju-shrestha/Git-Github.git
  06f329f..07fd994  master -> master
branch 'master' set up to track 'origin/master'.

User@DESKTOP-SVKOMKA MINGW64 ~/Downloads/Git-Github (master)
$ |
```

Now pull the changes made by collaborators.

```
User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (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 restore <file>..." to discard changes in working directory)
    modified:   Git & Github.docx

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ~$t & Github.docx
  ~WRL0005.tmp

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

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 327 bytes | 29.00 KiB/s, done.
From https://github.com/raju-shrestha/Git-Github
  06f329f..07fd994  master      -> origin/master
Updating 06f329f..07fd994
Fast-forward
 Update.txt | 3 ++
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-SVKOMKA MINGW64 ~/Desktop/Github (master)
```

Finally, last update is pulled and the repository is upto date.