

Git Commands

Git Commands with Explanations and Representations

Git is free and open-source software for distributed version control: tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

Try It:

`git --version`



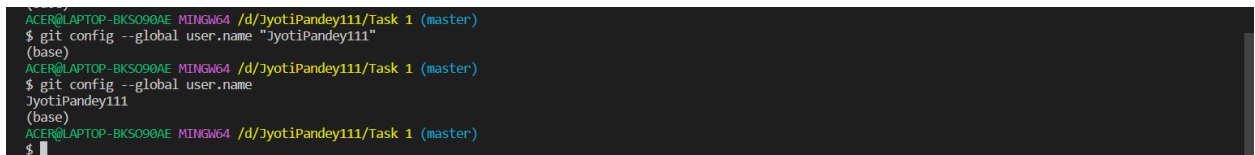
```
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1
$ git --version
git version 2.37.3.windows.1
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1
$
```

Configure Git: SETUP

1. Configuring user information used across all local repositories. It is a good idea to introduce yourself to Git with your name and public email address before doing any operation.

Try It:

`git config --global user.name "[firstname lastname]"`



```
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git config --global user.name "JyotiPandey111"
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git config --global user.name
JyotiPandey111
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$
```

set a name that is identifiable for credit when review version history

`git config --global user.email "[valid-email]"`



```
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git config --global user.email "jyotipandeydip@gmail.com"
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git config --global user.email
jyotipandeydip@gmail.com
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$
```

set an email address that will be associated with each history marker

2. SETUP & INIT: Open the terminal in your project and execute the below command.

Try It:

`git init`

```
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1
$ git init
Initialized empty Git repository in D:/JyotiPandey111/Task 1/.git/
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$
```

Initialize an existing directory as a Git repository

You've now initialized the working directory—you may notice a new directory created, named ".git".

3. To list changes that you've made but not yet added in staging or committed

Try It:

`git status`

```
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    git Commands.docx
    README.md
    Screenshots/
    ~$t Commands.docx

nothing added to commit but untracked files present (use "git add" to track)
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$
```

Files are untracked.

4. Next, tell Git to take a snapshot of the contents of all files under the current director

Try It:

`git add .`

```

ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git add .
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Git Commands.docx
    new file:   README.md
    new file:   Screenshots/Screenshot (1).png
    new file:   Screenshots/Screenshot (2).png
    new file:   Screenshots/Screenshot (3).png
    new file:   Screenshots/Screenshot (4).png
    new file:   Screenshots/Screenshot (6).png
    new file:   Screenshots/Screenshot (7).png
    new file:   ~$t Commands.docx

(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$

```

This snapshot is now stored in a temporary staging area which Git calls the "index". Here we have tracked all the file.

5. You can permanently store the contents of the index using the below command.

Try It:

`git commit -m "commit message"`

```

ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git commit -m "First Commit"
[master e0a2324] First Commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Screenshots/Screenshot (9).png
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git status
On branch master
nothing to commit, working tree clean
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$

```

Files went into staging.

6. Viewing project history

Try It:

`git log`

```

ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$ git log
commit 25e5de80c6f34a676476df356d3198adaebf711 (HEAD -> master)
Author: JyotiPandey111 <jyotipandeydip@gmail.com>
Date: Mon Oct 17 20:32:46 2022 +0530

    Updating Screenshots

commit e0a2324e31296476ba496407056ad63eeac25094
Author: JyotiPandey111 <jyotipandeydip@gmail.com>
Date: Mon Oct 17 20:27:30 2022 +0530

    First Commit

commit 43f64e4bfe578e417271905b611b62ec3ec4c11
Author: JyotiPandey111 <jyotipandeydip@gmail.com>
Date: Mon Oct 17 20:26:40 2022 +0530

    First Commit
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (master)
$

```

7. Remote add to origin

Try It:

git remote add origin [url of repository]

```
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git status
On branch main
nothing to commit, working tree clean
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git remote add origin https://github.com/yotiPandey111/Industry-Ready-Projects-Tasks.git
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git remote -v
origin https://github.com/yotiPandey111/Industry-Ready-Projects-Tasks.git (fetch)
origin https://github.com/yotiPandey111/Industry-Ready-Projects-Tasks.git (push)
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$
```

Managing Branches:

A single Git repository can maintain multiple branches of development.

1. To Rename a "master" branch to "main", use.

Try It:

git branch -M main

```
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (master)
$ git branch
* master
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (master)
$ git branch -M main
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git branch
* main
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$
```

2. To create a new branch named "branch_name", use.

Try It:

git branch branch_name

```
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git branch
* main
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git branch new
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$ git branch
* main
  new
(base)
ACER@LAPTOP-BKS09BAE MINGW64 /d/yotiPandey111/Task 1 (main)
$
```

3. To list all branches

Try It:

git branch

```
$ git branch
* main
  new
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (main)
$
```

Output: Default branch name “main” and newly created branch will be visible

4. To start working on another branch you can switch branches using the below command.

Try It:

`git switch branch_name`

```
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (main)
$ git branch
* main
  new
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (main)
$ git switch new
Switched to branch 'new'
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$ git branch
main
* new
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$
```

5. To delete the branch.

Try It:

`git branch -d branch_name`

```
$ git branch
* main
  new1
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$ git branch -d new1
Deleted branch new1 (was a51bacf).
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$ git branch
main
* new
(base)
ACER@LAPTOP-BKS090AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$
```

Note: This command ensures that the changes in the experimental branch are already in the current branch.

Using Git for Collaboration

1. Clone a repo

Try It:

`git clone repo_url`

```

(base)
ACER@LAPTOP-BKSO90AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$ git clone https://github.com/JyotiPandey111/Tutorial-for-Git---iNeuron-2022.git
Cloning into 'Tutorial-for-Git---iNeuron-2022' ...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 24 (delta 5), reused 21 (delta 2), pack-reused 0
Receiving objects: 100% (24/24), done.
Resolving deltas: 100% (5/5), done.
(base)
ACER@LAPTOP-BKSO90AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$

```

2. Push a repo

Try It:

`git push origin "branch_name"`

```

ACER@LAPTOP-BKSO90AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$ git push origin new
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 80.41 KiB | 3.09 MiB/s, done.
Total 10 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 3 local objects.
To https://github.com/JyotiPandey111/Industry-Ready-Projects-Tasks.git
   bd75eb1..47750ef  new -> new
(base)
ACER@LAPTOP-BKSO90AE MINGW64 /d/JyotiPandey111/Task 1 (new)
$

```