

```
admin@DESKTOP-EOD6V01 MINGW64 ~
$ cd sam

admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ git init
Reinitialized existing Git repository in C:/Users/admin/sam/.git/
admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ gti add .
bash: gti: command not found

admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ git add .

admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ touch file.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ git log
commit 22d13179a937c08b96d697f87eeb6f9127cfde87 (HEAD -> feature-branch, master)
Author: pooja <poojadigu@gmail.com>
Date:   Tue Oct 14 08:50:29 2025 +0100
```

File created(v1)

```
admin@DESKTOP-EOD6V01 MINGW64 ~/sam (feature-branch)
$ git checkout -m master
Switched to branch 'master'
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/sam (master)
$ git add .
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/sam (master)
$ git commit -m "sonu git is easy"
[master 3ed1de6] sonu git is easy
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project (master)
```

```
$ mkdir ziyu
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project (master)
```

```
$ cd ziyu
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git init
```

```
Initialized empty Git repository in C:/Git project/ziyu/.git/
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ touch ziyu file.txt
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git add .
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git commit -m "Initial commit - adding a new file"
```

```
[master (root-commit) ac3ab56] Initial commit - adding a new file
```

```
 2 files changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 file.txt
```

```
create mode 100644 ziyu
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git branch feature-branch
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git checkout -b feature-branch
```

```
fatal: a branch named 'feature-branch' already exists
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git checkout master
```

```
Already on 'master'
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git merge feature-branch
```

```
Already up to date.
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git checkout master
```

```
Already on 'master'
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git add
```

```
Nothing specified, nothing added.
```

```
hint: Maybe you wanted to say 'git add .'?
```

```
hint: Disable this message with "git config set advice.addEmptyPathspec false"
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ git branch -d feature-branch
```

```
Deleted branch feature-branch (was ac3ab56).
```

```
admin@DESKTOP-EOD6V01 MINGW64 /c/Git project/ziyu (master)
```

```
$ |
```

```
◆ MINGW64/c/Users/admin/Desktop/stah_demo
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop
$ mkdir stah_demo

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop (master)
$ cd stah_demo

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git init
Initialized empty Git repository in C:/Users/admin/Desktop/stah_demo/.git/
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ touch main.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git add main.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ echo "print('Hello world')" >main.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git commit -m "Your first commit"
[master (root-commit) cf62ca4] Your first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ echo "print('This is a stash demo')" >main.txt

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git status
On branch 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:   main.txt

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

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git stash save "your stash message"
warning: in the working copy of 'main.txt', LF will be replaced by CRLF the next time
Saved working directory and index state On master: your stash message

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git stash list
stash@{0}: On master: your stash message

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (master)
$ git checkout -b target-branch
Switched to a new branch 'target-branch'

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stah_demo (target-branch)
$ git stash apply
On branch target-branch
```

```
MINGW64:/c/Users/admin/Desktop/stash_demo/stash_demo
$ git checkout -b target_branch
Switched to a new branch 'target_branch'

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash apply
On branch target_branch
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:   main.txt

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

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git add .

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash -m "Add input entry validation feature"
Saved working directory and index state on target_branch: Add input entry validation feature

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash list
stash@{0}: on target_branch: Add input entry validation feature
stash@{1}: on master: your stash message

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash show stash@{0}
main.txt | 2 ++
1 file changed, 1 insertion(+), 1 deletion(-)

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash apply stash@{1}
On branch target_branch
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:   main.txt

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

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git add .

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash pop
On branch target_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   main.txt

Dropped refs/stash@{0} (877f81fdcaa7d498b6efcc40e8532210ed6d49c4)

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash list
stash@{0}: on master: your stash message

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash push -m "say my name"
Saved working directory and index state on target_branch: say my name

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ git stash list
stash@{0}: On target_branch: say my name
stash@{1}: on master: your stash message

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/stash_demo/stash_demo (target_branch)
$ |
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/ntg (master)
$ cd "C:\Users\admin\Desktop\ntg"

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/ntg (master)
$ git clone "https://github.com/Harshithalokesh385/program_4.git"
Cloning into 'program_4'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.

admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/ntg (master)
$ _
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
$ cd "C:\Users\admin\Desktop\p5"
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5
$ git clone "https://github.com/archanang17-png/archana-p5-.git"
fatal: destination path 'archana-p5-' already exists and is not an empty directory.
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5
$ cd "C:\Users\admin\Desktop\p5\archana-p5-"
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
$ git fetch
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
$ git rebase origin/main
Current branch main is up to date.
```

```
admin@DESKTOP-E0D6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
$ |
```

noop

```
# Rebase e460578..e460578 onto e460578 (1 command)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#       create a merge commit using the original merge commit's
#       message (or the oneline, if no original merge commit was
#       specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                   to this position in the new commits. The <ref> is
#                   updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
```

Successfully rebased and updated refs/heads/main.

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ git rebase --abort  
fatal: no rebase in progress
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ git push origin main --force  
Everything up-to-date
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ git rebase --continue  
fatal: no rebase in progress
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ git status  
On branch main  
Your branch is up to date with 'origin/main'.
```

nothing to commit, working tree clean

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ rm -rf .git/rebase-merge
```

```
admin@DESKTOP-EOD6V01 MINGW64 ~/Desktop/p5/archana-p5- (main)
```

```
$ git status  
On branch main  
Your branch is up to date with 'origin/main'.
```

nothing to commit, working tree clean