

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
```

```
$ cd zeeshan
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git init
```

```
Reinitialized existing Git repository in C:/Users/admin/zeeshan/.git/
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git branch new
```

```
fatal: not a valid object name: 'master'
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ vim new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git add new.c
```

```
warning: in the working copy of 'new.c', LF will be replaced by CRLF the next time
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git commit -m "new"
```

```
[master (root-commit) 866dd29] new
```

```
1 file changed, 5 insertions(+)
```

```
create mode 100644 new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git log
```

```
commit 866dd2952cdbc90c1390313ca19764f1d3a36da4 (HEAD -> master)
```

```
Author: Zeeshan <zenaftab02@gmail.com>
```

```
Date: Tue Mar 19 14:54:46 2024 +0530
```

```
new
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git branch new
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git checkout new
```

```
Switched to branch 'new'
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ vim new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ vim new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ git add new.c
```

```
warning: in the working copy of 'new.c', LF will be replaced by CRLF the next time  
git touches it
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ vim hello.txt
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ git add hello.txt
```

```
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next  
time git touches it
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
A      hello.txt
```

```
M      new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ ls
```

```
hello.txt  new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git new.c
```

```
git: 'new.c' is not a git command. See 'git --help'.
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ vim new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git checkout new
```

```
Switched to branch 'new'
```

```
A      hello.txt
```

```
M      new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ vim new.c
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new)
```

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
A      hello.txt
```

```
M      new.c
```



Switched to branch 'master'

A hello.txt

M new.c

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ vim new.c

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ git merge new

Already up to date.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: hello.txt

modified: new.c

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.c

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ git branch new1

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: hello.txt

modified: new.c

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.c

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ vim sa.txt

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)

\$ git add sa.txt

warning: in the working copy of 'sa.txt', LF will be replaced by CRLF the next time

git touches it

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git checkout new1
Switched to branch 'new1'
A       hello.txt
M       new.c
A       sa.txt

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git log --oneline
866dd29 (HEAD -> new1, new, master) new

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git stash
warning: in the working copy of 'new.c', LF will be replaced by CRLF the next time
Git touches it
Saved working directory and index state WIP on new1: 866dd29 new

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git stash list
stash@{0}: WIP on new1: 866dd29 new

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git stash show stash@{1}
fatal: log for 'stash' only has 1 entries

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git stash apply
On branch new1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hello.txt
    new file:   sa.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.c

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git stash pop
error: Your local changes to the following files would be overwritten by merge:
    new.c
Please commit your changes or stash them before you merge.
Aborting
on branch new1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
```



on branch new1

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: hello.txt

new file: sa.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.c

The stash entry is kept in case you need it again.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git stash list

stash@{0}: WIP on new1: 866dd29 new

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git clone https://zeeshanaf02.github.io/getyourwords.io/

Cloning into 'getyourwords.io'...

fatal: repository 'https://zeeshanaf02.github.io/getyourwords.io/' not found

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git clone https://github.com/zeeshanaf02/getyourwords.io.git

Cloning into 'getyourwords.io'...

remote: Enumerating objects: 26, done.

remote: Counting objects: 100% (26/26), done.

remote: Compressing objects: 100% (24/24), done.

remote: Total 26 (delta 14), reused 6 (delta 2), pack-reused 0

Receiving objects: 100% (26/26), 10.90 KiB | 10.90 MiB/s, done.

Resolving deltas: 100% (14/14), done.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git fetch origin

fatal: 'origin' does not appear to be a git repository

fatal: could not read from remote repository.

Please make sure you have the correct access rights  
and the repository exists.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git fetch https://github.com/zeeshanaf02/getyourwords.io.git

remote: Enumerating objects: 26, done.

remote: Counting objects: 100% (26/26), done.

remote: Compressing objects: 100% (24/24), done.

remote: Total 26 (delta 14), reused 6 (delta 2), pack-reused 0

Unpacking objects: 100% (26/26), 10.88 KiB | 67.00 KiB/s, done.

From https://github.com/zeeshanaf02/getyourwords.io

\* branch HEAD -> FETCH\_HEAD



From https://github.com/zeeshanaf02/getyourwords.io  
\* branch HEAD -> FETCH\_HEAD

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git checkout new1

Already on 'new1'

A hello.txt

M new.c

A sa.txt

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git rebase master new1

error: cannot rebase: You have unstaged changes.

error: additionally, your index contains uncommitted changes.

error: Please commit or stash them.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git rebase https://github.com/zeeshanaf02/getyourwords.io.git new1

fatal: invalid upstream 'https://github.com/zeeshanaf02/getyourwords.io.git'

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git commit -m "last"

[new1 342303e] last

2 files changed, 3 insertions(+)

create mode 100644 hello.txt

create mode 100644 sa.txt

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git rebase master new1

error: cannot rebase: You have unstaged changes.

error: Please commit or stash them.

admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)

\$ git add .

warning: adding embedded git repository: getyourwords.io

hint: You've added another git repository inside your current repository.

hint: Clones of the outer repository will not contain the contents of

hint: the embedded repository and will not know how to obtain it.

hint: If you meant to add a submodule, use:

hint:

hint: git submodule add <url> getyourwords.io

hint:

hint: If you added this path by mistake, you can remove it from the

hint: index with:

hint:

hint: git rm --cached getyourwords.io

hint:

hint: See "git help submodule" for more information.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git commit -m "hhh"
[new1 c749f82] hhh
2 files changed, 3 insertions(+)
create mode 160000 getyourwords.io
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git rebase master new1
Current branch new1 is up to date.
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git add .
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git rebase --continue
fatal: No rebase in progress?
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git push -f origin new1
fatal: 'origin' does not appear to be a git repository
fatal: could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git push -f master new1
fatal: 'master' does not appear to be a git repository
fatal: could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git push -f https://github.com/zeeshanaf02/getyourwords.io.git new1
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 906 bytes | 906.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'new1' on GitHub by visiting:
remote:   https://github.com/zeeshanaf02/getyourwords.io/pull/new/new1
remote:
```



```
remote:
To https://github.com/zeeshanaf02/getyourwords.io.git
* [new branch]      new1 -> new1
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (new1)
$ git checkout master
warning: unable to rmdir 'getyourwords.io': Directory not empty
Switched to branch 'master'
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git merge new
Already up to date.
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git commit -m "my message"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    getyourwords.io/
```

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

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git add .
warning: adding embedded git repository: getyourwords.io
hint: You've added another git repository inside your current repository.
hint: clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> getyourwords.io
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached getyourwords.io
hint:
hint: See "git help submodule" for more information.
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git commit -m "my message"
[master 38be1aa] my message
1 file changed, 1 insertion(+)
create mode 160000 getyourwords.io
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git tag v1.0
```



```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git tag v1.0
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git tag  
v1.0
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git tag -a v1.1 -m "tag for release ver 1.1"
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git show v1.0
```

```
commit 38be1aa965161f51452366c0d8a0f12569bda869 (HEAD -> master, tag: v1.1, tag: v1.0)
```

```
Author: Zeeshan <zenaftab02@gmail.com>
```

```
Date: Tue Mar 19 15:49:58 2024 +0530
```

```
my message
```

```
diff --git a/getyourwords.io b/getyourwords.io
```

```
new file mode 160000
```

```
index 0000000..3381d22
```

```
--- /dev/null
```

```
+++ b/getyourwords.io
```

```
@@ -0,0 +1 @@
```

```
+Subproject commit 3381d2230b9acca7cd19166628f4ddda97615ca0
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git tag -l "v1*"
```

```
error: unknown switch `l'
```

```
usage: git tag [-a | -s | -u <key-id>] [-f] [-m <msg> | -F <file>] [-e]  
          <tagname> [<commit> | <object>]
```

```
or: git tag -d <tagname>...
```

```
or: git tag [-n<num>] -l [--contains <commit>] [--no-contains <commit>]  
          [--points-at <object>] [--column[=<options>] | --no-column]  
          [--create-reflog] [--sort=<key>] [--format=<format>]  
          [--merged <commit>] [--no-merged <commit>] [<pattern>...]
```

```
or: git tag -v [--format=<format>] <tagname>...
```

-l, --list	list tag names
-n[<n>]	print <n> lines of each tag message
-d, --delete	delete tags
-v, --verify	verify tags

Tag creation options

-a, --[no-]annotate annotated tag, needs a message

-m, --message <message>  
tag message

```

-m, --message <message>      tag message
-F, --[no-]file <file>       read message from file
-e, --[no-]edit               force edit of tag message
-s, --[no-]sign               annotated and GPG-signed tag
--[no-]cleanup <mode>        how to strip spaces and #comments from message
-u, --[no-]local-user <key-id> use another key to sign the tag
-f, --[no-]force              replace the tag if exists
--[no-]create-reflog          create a reflog

```

## Tag listing options

```

--[no-]column[=<style>]      show tag list in columns
--contains <commit>          print only tags that contain the commit
--no-contains <commit>       print only tags that don't contain the commit
--merged <commit>            print only tags that are merged
--no-merged <commit>         print only tags that are not merged
--[no-]omit-empty            do not output a newline after empty formatted refs
--[no-]sort <key>            field name to sort on
--[no-]points-at <object>    print only tags of the object
--[no-]format <format>       format to use for the output
--[no-]color[=<when>]        respect format colors
-i, --[no-]ignore-case       sorting and filtering are case insensitive

```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git push oringin v1.0
```

```
fatal: 'oringin' does not appear to be a git repository
```

```
fatal: could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```

```
$ git push origin v1.0
```

```
fatal: 'origin' does not appear to be a git repository
```

```
fatal: could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
```



```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git push master v1.0
fatal: 'master' does not appear to be a git repository
fatal: could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ git push https://github.com/zeeshanaf02/themomo.github.io.git v1.0
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 499 bytes | 499.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/zeeshanaf02/themomo.github.io.git
 * [new tag]          v1.0 -> v1.0
```

```
admin@DESKTOP-FFTI550 MINGW64 ~/zeeshan (master)
$ cd ..
```

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
$ cd D
bash: cd: D: No such file or directory
```

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
$ cd \D
bash: cd: D: No such file or directory
```

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
$ cd D:\zeeshan\Git cherry picking
bash: cd: too many arguments
```

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
$ cd \D
bash: cd: D: No such file or directory
```

```
admin@DESKTOP-FFTI550 MINGW64 ~ (master)
$ cd /D
```

```
admin@DESKTOP-FFTI550 MINGW64 /D
$ cd zeeshan
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan
$ cd Git cherry picking
bash: cd: too many arguments
```

bash: cd: too many arguments

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan
```

```
$ cd "
```

```
> ^C
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan
```

```
$ cd "Git cherry picking"
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking
```

```
$ git init
```

```
Initialized empty Git repository in D:/Zeeshan/Git cherry picking/.git/
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ vim hhe.txt
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ git add .|git commit -m "one"
```

```
warning: in the working copy of 'hhe.txt', LF will be replaced by CRLF the next time Git touches it
```

```
[master (root-commit) 9ba518b] one
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 hhe.txt
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ vim hee.txt
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ git add .|git commit -m "two"
```

```
warning: in the working copy of 'hee.txt', LF will be replaced by CRLF the next time Git touches it
```

```
fatal: Unable to create 'D:/Zeeshan/Git cherry picking/.git/index.lock': File exists.
```

Another git process seems to be running in this repository, e.g. an editor opened by 'git commit'. Please make sure all processes are terminated then try again. If it still fails, a git process may have crashed in this repository earlier: remove the file manually to continue.

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ vim the.c
```

```
admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
```

```
$ git add .|git commit -m "three"
```



```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git add .|git commit -m "three"
warning: in the working copy of 'the.c', LF will be replaced by CRLF the next time
git touches it
[master 7a2daf1] three
2 files changed, 2 insertions(+)
create mode 100644 hee.txt
create mode 100644 the.c
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git reflog
7a2daf1 (HEAD -> master) HEAD@{0}: commit: three
9ba518b HEAD@{1}: commit (initial): one
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git add .|git commit -m "one"
On branch master
nothing to commit, working tree clean
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git add .|git commit -m "two"
On branch master
nothing to commit, working tree clean
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git add .|git commit -m "three"
On branch master
nothing to commit, working tree clean
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git reflog
7a2daf1 (HEAD -> master) HEAD@{0}: commit: three
9ba518b HEAD@{1}: commit (initial): one
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git 7a2daf1
git: '7a2daf1' is not a git command. See 'git --help'.
```

```
admin@DESKTOP-FFT1550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git cherry-pick 7a2daf1
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:
```

```
git commit --allow-empty
```

```
otherwise, please use 'git cherry-pick --skip'
On branch master
```



otherwise, please use "git cherry-pick --skip

on branch master

You are currently cherry-picking commit 7a2daf1.

(all conflicts fixed: run "git cherry-pick --continue")

(use "git cherry-pick --skip" to skip this patch)

(use "git cherry-pick --abort" to cancel the cherry-pick operation)

nothing to commit, working tree clean

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)

\$ git log

commit 7a2daf161f7232b5542cc650f31acf1c74c148fd (HEAD -> master)

Author: Zeeshan <zenaftab02@gmail.com>

Date: Tue Mar 19 16:24:37 2024 +0530

three

commit 9ba518bc1349c6a4785982690058b2535a40d518

Author: Zeeshan <zenaftab02@gmail.com>

Date: Tue Mar 19 16:13:14 2024 +0530

one

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)

\$ git commitID

git: 'commitID' is not a git command. See 'git --help'.

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)

\$ git show commitID

fatal: ambiguous argument 'commitID': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

'git <command> [<revision>...] -- [<file>...]'

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)

\$ git show 9ba518bc1349c6a4785982690058b2535a40d518

commit 9ba518bc1349c6a4785982690058b2535a40d518

Author: Zeeshan <zenaftab02@gmail.com>

Date: Tue Mar 19 16:13:14 2024 +0530

one

diff --git a/hhe.txt b/hhe.txt

new file mode 100644



```
new file mode 100644
index 0000000..73bc95c
--- /dev/null
+++ b/hhe.txt
@@ -0,0 +1 @@
+lsjjfnvnsfjjfj

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git log -n 5
commit 7a2daf161f7232b5542cc650f31acf1c74c148fd (HEAD -> master)
Author: Zeeshan <zenaftab02@gmail.com>
Date: Tue Mar 19 16:24:37 2024 +0530

    three

commit 9ba518bc1349c6a4785982690058b2535a40d518
Author: Zeeshan <zenaftab02@gmail.com>
Date: Tue Mar 19 16:13:14 2024 +0530

    one

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git revert "abs123"
fatal: bad revision 'abs123'

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git revert "abc123"
fatal: bad revision 'abc123'

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git log --oneline
7a2daf1 (HEAD -> master) three
9ba518b one

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git reset -hard head.1
error: did you mean '--hard' (with two dashes)?

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git reset --hard head.1
fatal: ambiguous argument 'head.1': unknown revision or path not in the working tr
```

```
)
$ git reset --hard head.1
fatal: ambiguous argument 'head.1': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master|CHERRY-PICKING)
$ git revert "7a2daf1"
[master 6735feb] Revert "three"
Date: Tue Mar 19 16:24:37 2024 +0530
2 files changed, 2 deletions(-)
delete mode 100644 hee.txt
delete mode 100644 the.c

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git log -oneline
fatal: unrecognized argument: -oneline

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git log --oneline
6735feb (HEAD -> master) Revert "three"
7a2daf1 three
9ba518b one

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git reset -hard head.1
error: did you mean '--hard' (with two dashes)?

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
$ git reset --hard head.1
fatal: ambiguous argument 'head.1': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

admin@DESKTOP-FFTI550 MINGW64 /D/zeeshan/Git cherry picking (master)
$
```