

# GIT Commands

## 1. Initializing a repository :

```
divum@divum-Latitude-7490:~$ cd Documents
divum@divum-Latitude-7490:~/Documents$ cd Intern
divum@divum-Latitude-7490:~/Documents/Intern$ mkdir git2
divum@divum-Latitude-7490:~/Documents/Intern$ cd git2
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git init
Initialized empty Git repository in /home/divum/Documents/Intern/git2/.git/
```

## 2. Staging files and Viewing the status:

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ touch file.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git status
On branch master

No commits yet

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

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

## 3. Committing the staged files and Viewing the status :

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git add file.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git commit -m"text file committed"
[master (root-commit) d7d3561] text file committed
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git status
On branch master
nothing to commit, working tree clean
```

## 4. Skipping the staging area :

Only tracked files can skip the staging area, to add your file to tracker type “git add --a” or “git add filename.extension”.

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ touch file3.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git commit -a -m"skipping the staging"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3.txt

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

Now let's skip the staging by typing “git commit -a -m “Commit Message””

Now if we do “git status” it will show that the working directory is clean.

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git add file3.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git commit -m"file3 committed"
[master 89d60c0] file3 committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git commit -a -m"skipping the staging"
On branch master
nothing to commit, working tree clean
```

#### 5.Removing files :

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git rm file2.txt
rm 'file2.txt'
divum@divum-Latitude-7490:~/Documents/Intern/git2$ ls
file3.txt  file.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git commit -m"deleted the file2 from git repo"
[master 87a98bc] deleted the file2 from git repo
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 file2.txt
```

#### 6.Renaming or moving files :

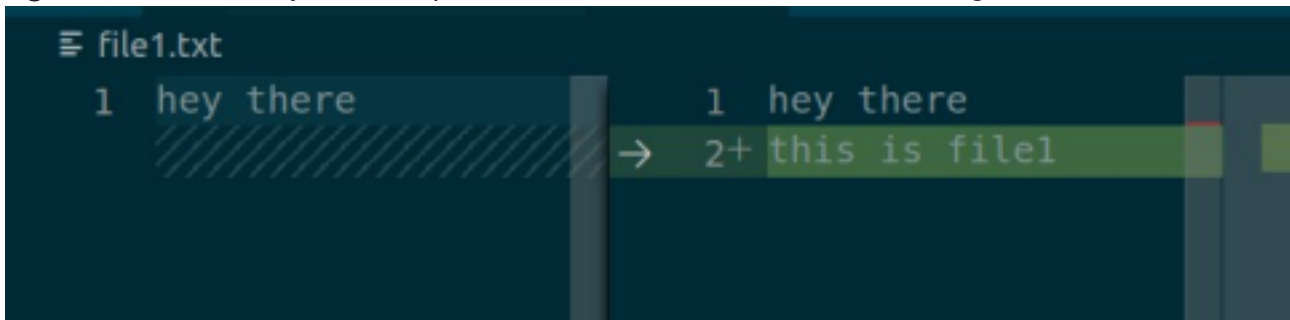
```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git mv file3.txt renamedfile.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$ ls
file.txt  renamedfile.txt
```

#### 7.Viewing the staged/unstaged changes :

**\$ git diff** # Diffing is a function that takes two input data sets and outputs the changes between them. git diff is a multi-use Git command that when executed runs a diff function on Git data sources.

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git diff
diff --git a/renamedfile.txt b/renamedfile.txt
index e69de29..c750817 100644
--- a/renamedfile.txt
+++ b/renamedfile.txt
@@ -0,0 +1,2 @@
+Hey there
+Its me
\ No newline at end of file
```

**\$ git difftool** # allows you to compare and edit files between revisions using common diff tools.



```

≡ file1.txt
1  hey there
→ 2+ this is file1

```

#### 8.Viewing the history :

git log # Full history

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git log
commit b3c5beb4f6c96d0cbe69f707113389084ae2f434 (HEAD -> master)
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 11:03:45 2022 +0530

    file got modified

commit 87a98bc61c4848cbad8c5d0e7c5071d4158b3e0c
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:55:58 2022 +0530

    deleted the file2 from git repo

commit 89d60c05a0f78ac80d1465e0ea5d450c229903ef
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:49:25 2022 +0530

    file3 committed

commit 0b83fc3442ddd32ef0c191fa2919dbe1c595f6ed
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:41:53 2022 +0530

    File-2 committed
```

git log --oneline # Summary

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git log --oneline
b3c5beb (HEAD -> master) file got modified
87a98bc deleted the file2 from git repo
89d60c0 file3 committed
0b83fc3 File-2 committed
d7d3561 text file committed
divum@divum-Latitude-7490:~/Documents/Intern/git2$
```

git log --reverse # Lists the commits from the oldest to the newest

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git log --reverse
commit d7d3561d70b17301d3002f60b1b0e01100ebc
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:59:13 2022 +0530

    text file committed

commit 0b83fc3442ddd32ef0c191fa2919dbe1c595f6ed
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:41:53 2022 +0530

    File-2 committed

commit 89d60c05a0f78ac80d1465e0ea5d450c229903ef
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:49:25 2022 +0530

    file3 committed

commit 87a98bc61c4848cbad8c5d0e7c5071d4158b3e0c
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:55:58 2022 +0530

    deleted the file2 from git repo
(....skipping....)
commit b3c5beb4f6c96d0cbe69f707113389084ae2f434
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 11:03:45 2022 +0530

    text file committed

commit 0b83fc3442ddd32ef0c191fa2919dbe1c595f6ed
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:41:53 2022 +0530

    File-2 committed

commit 89d60c05a0f78ac80d1465e0ea5d450c229903ef
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:49:25 2022 +0530

    file3 committed

commit 87a98bc61c4848cbad8c5d0e7c5071d4158b3e0c
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:55:58 2022 +0530

    deleted the file2 from git repo

commit b3c5beb4f6c96d0cbe69f707113389084ae2f434 (HEAD -> master)
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 11:03:45 2022 +0530

    file got modified
```

9. Viewing a commit :

git show commit\_ID # Shows the given commit

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git show 89d60c0
commit 89d60c05a0f78ac80d1465e0ea5d450c229903ef
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:49:25 2022 +0530

    file3 committed

diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..e69de29
```

git show HEAD # Shows the last commit

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git show HEAD
commit b3c5beb4f6c96d0cbe69f707113389084ae2f434 (HEAD -> master)
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 11:03:45 2022 +0530

    file got modified

diff --git a/file.txt b/file.txt
index e69de29..6acb7b9 100644
--- a/file.txt
+++ b/file.txt
@@ -0,0 +1,2 @@
+hey there this is 2nd docu
+git assgnt 2
\ No newline at end of file
diff --git a/file3.txt b/renamedfile.txt
similarity index 100%
rename from file3.txt
rename to renamedfile.txt
divum@divum-Latitude-7490:~/Documents/Intern/git2$
```

git show HEAD~2 # Two steps before the last commit

```
divum@divum-Latitude-7490:~/Documents/Intern/git2$ git show HEAD~2
commit 89d60c05a0f78ac80d1465e0ea5d450c229903ef
Author: Sailochana <sailochana@divum.in>
Date:   Wed Aug 24 10:49:25 2022 +0530

    file3 committed

diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..e69de29
```

git show HEAD:file\_nm.ext# Shows the version of file.js stored in the last commit



10.Unstaging files (undoing git add) :

git restore --staged file\_nm.ext# Copies the last version of file.js from repo to index





### 11. Discarding local changes :

`git restore file_nm.ext` # Copies file.js from index to working directory



`git restore file1.txt file2.txt` # Restores multiple files in working directory  
`git restore .` # Discards all local changes (except untracked files)

`git clean -fd` # Removes all untracked files



### 12. Restoring an earlier version of a file :

`git restore --source=HEAD~2 file_nm.ext`



### 13.Introduction:

Git is a version control system used for tracking changes in computer files. It is generally used for source code management in software development. Git is used to tracking changes in the source code. The distributed version control tool is used for source code management.

### 14.Getting a repository:

The local repository is a Git repository that is stored on your computer. The remote repository is a Git repository that is stored on some remote computer.

### 15.Viewing the History :

`git log --stat # Shows the list of modified files`



`git log --patch # Shows the actual changes (patches)`



#### 16. Filtering the History:

```
git log -3 # Shows the last 3 entries
git log --author="Mosh"
git log --before="2020-08-17"
git log --after="one week ago"
git log --grep="GUI" # Commits with "GUI" in their message
git log -S"GUI" # Commits with "GUI" in their patches
git log hash1..hash2 # Range of commits
git log file.txt # Commits that touched file.txt
```

#### 17. Formatting the Log output:

```
git log --pretty=format:"%an committed %H"
#<full-commit-message> reference. <abbrev-hash> (<title-line>, <short-author-date>) This format
is used to refer to another commit in a commit message and is the same as --
pretty='format:%C(auto)%h (%s, %ad)'
```



#### 18.alias:

```
git config --global alias.lg "log --oneline"
```



#### 19.Viewing Changes across the Commit:



#### 20.Checking out a commit:



#### 21.Finding bugs using bisect:

The git bisect command is used to discover the commit that has introduced a bug in the code.It helps track down the commit where the code works and the commit where it does not, hence, tracking down the commit that introduced the bug into the code.



#### 22.Finding the contributor using shortlog:

The git shortlog command is a special version of git log intended for creating release announcements.



#### 23.Viewing the History of the File:

`git log`

#### 24.Restoring the Deleted file:

`git restore <file_nm>` #Git provides ways to recover a deleted file at any point in this life cycle of changes. If you have not staged the deletion yet, simply run ``git restore <filename>``



and the file will be restored from the index.



## 25. Finding the author of line using blame:

The git blame command is used to examine the contents of a file line by line and see when each line was last modified and who the author of the modifications was.



## 26. Tagging:

git tag v1.0 # Tags the last commit as v1.0

git tag v1.0 5e7a828 # Tags an earlier commit

git tag # Lists all the tags

git tag -d v1.0 # Deletes the given tag



## 27. Browsing History using VS Code.:

Open the file to view the history, and then Press F1 and select/type "Git: View History", "Git: View File History" or "Git: View Line History"