

# Version Control System setup and usage using GIT

What is version control ?

Version control systems are a category of software tools that help a software team manage changes to source code over time.

## 1. Setting up a repository

Command 1 : `git init`

Purpose of Command : Create an empty local repository.

Usages : `git init`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/git$ ls -a
.  ..
abhishek@abhishek:~/Abhishek/fosslab/git$ git init
Initialized empty Git repository in /home/abhishek/Abhishek/fosslab/git/.git/
abhishek@abhishek:~/Abhishek/fosslab/git$ ls -a
.  ..  .git
abhishek@abhishek:~/Abhishek/fosslab/git$
```

Command 2 : `git config`

Purpose of Command : Configure the author name and email address to be used with your commits.

Usages : `git config --global user.name <username>`  
`git config --global user.email <email ID>`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/git$ git config --global user.mail abhishekmanoharan@cet.ac.in
abhishek@abhishek:~/Abhishek/fosslab/git$ git config --global user.name abhishekmanoharan
abhishek@abhishek:~/Abhishek/fosslab/git$
```

Command 3 : `git remote add`

Purpose of Command : To add a remote repo url to your local `git config`

Usages : `git remote add <remote_name><remote_repo_url>`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/git$ git remote add origin https://github.com/abhishekmanoharan/git_basics.git
abhishek@abhishek:~/Abhishek/fosslab/git$
```

## 2 Adding content

Command 4 : `git add`

Purpose of Command : Add one or more files to staging.

Usages : `git add <filename>` #to add a particular file  
`git add .` #to add all files in the directory

Sample Input/Output :

```
File Edit View Search Terminal Help
abhishek@abhishek:~/Abhishek/fosslab/git$ touch hello
abhishek@abhishek:~/Abhishek/fosslab/git$ touch basics
abhishek@abhishek:~/Abhishek/fosslab/git$ touch new
abhishek@abhishek:~/Abhishek/fosslab/git$ git add hello
abhishek@abhishek:~/Abhishek/fosslab/git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   hello

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

        basics
        new

abhishek@abhishek:~/Abhishek/fosslab/git$ git add .
abhishek@abhishek:~/Abhishek/fosslab/git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   basics
        new file:   hello
        new file:   new

abhishek@abhishek:~/Abhishek/fosslab/git$
```

## 3 Committing the data

Command 5 : `git commit`

Purpose of Command : Commit changes to head (but not yet to the remote repository).

Usages : `git commit -m <commit message>`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/git$ git commit -m "Added 3 files"
[master (root-commit) 2dd26d9] Added 3 files
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 basics
create mode 100644 hello
create mode 100644 new
abhishek@abhishek:~/Abhishek/fosslab/git$ git status
On branch master
nothing to commit, working tree clean
abhishek@abhishek:~/Abhishek/fosslab/git$
```

## 4 Updating the local copy

Command 6 : `git push`

Purpose of Command : Send changes to the master branch of your remote repository

Usages : `git push <remote> <branch>`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/git$ git push -u origin master
Username for 'https://github.com': abhishekmanoharan
Password for 'https://abhishekmanoharan@github.com':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/abhishekmanoharan/git_basics.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
abhishek@abhishek:~/Abhishek/fosslab/git$
```

## 5 Checking out a repository

Command 7 : `git clone`

Purpose of Command : Create a working copy of a local repository.

Usages : `git clone <path>`

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fosslab/new_git$ git clone /Abhishek/fosslab/git
fatal: repository '/Abhishek/fosslab/git' does not exist
abhishek@abhishek:~/Abhishek/fosslab/new_git$ git clone ~/Abhishek/fosslab/git
Cloning into 'git'...
done.
abhishek@abhishek:~/Abhishek/fosslab/new_git$ ls
git
abhishek@abhishek:~/Abhishek/fosslab/new_git$
```

Command 8 :git checkout

Purpose of Command :To do basic branch operation.

Usages :git checkout -b <branch\_name> #create and switch  
git checkout<branch\_name> #switch to a branch  
git checkout -- <filename> #undo changes

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fossilab/git$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
abhishek@abhishek:~/Abhishek/fossilab/git$ git checkout -b trial
Switched to a new branch 'trial'
abhishek@abhishek:~/Abhishek/fossilab/git$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
abhishek@abhishek:~/Abhishek/fossilab/git$
```

## 6 Comparing different revisions

Command 9 :git diff / git show

Purpose of Command :Diffing is a function that takes two input data sets and outputs the changes between them

Usages :git diff <commit ID 1> <commit ID 2>

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fossilab/git$ git show
commit 2c1354ef972931b989cb4fdf4db1ab3b9fae2a4e (HEAD -> master, origin/master)
Author: abhishekmanoharan <abhishekmanoharan1999@gmail.com>
Date: Mon Feb 18 20:27:01 2019 +0530

    error happend

diff --git a/error b/error
new file mode 100644
index 0000000..e69de29
abhishek@abhishek:~/Abhishek/fossilab/git$ git diff 2dd26d9 2c1354e
diff --git a/error b/error
new file mode 100644
index 0000000..e69de29
```

## 7 Revert

Command 10 :git revert

Purpose of Command :The “git revert” command can be considered an 'undo' command.

Usages :git revert <HEAD>

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fossilab/git$ git log
commit 2c1354ef972931b989cb4fdf4db1ab3b9fae2a4e (HEAD -> master, origin/master)
Author: abhishekmanoharan <abhishekmanoharan1999@gmail.com>
Date: Mon Feb 18 20:27:01 2019 +0530

    error happend

commit 2dd26d97d5c311e441dbbee20e807f022550b6a2 (trial)
Author: abhishekmanoharan <abhishekmanoharan1999@gmail.com>
Date: Mon Feb 18 19:48:24 2019 +0530

    Added 3 files
abhishek@abhishek:~/Abhishek/fossilab/git$ git revert HEAD
[master 144959a] Revert "error happend"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 error
abhishek@abhishek:~/Abhishek/fossilab/git$ ls
basics hello new
```

GNU nano 2.9.3 /home/abhishek/Abhishek/fossilab/git/.git/COMMIT\_EDITMSG

Revert "error happend"

This reverts commit 2c1354ef972931b989cb4fdf4db1ab3b9fae2a4e.

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   deleted:    error
#
```

## 8 Conflict and conflict resolution

Merge conflicts occur when competing changes are made to the same line of a file, or when one person edits a file and another person deletes the same file.

Command 11 :git merge

Purpose of Command :The git merge command's primary responsibility is to combine separate branches and resolve any conflicting edits.

Usages :

Sample Input/Output :

```
abhishek@abhishek:~/Abhishek/fossilab/git$ git merge
Already up to date.
```

here there is no conflict since only one user is working in the repository. But if a conflict happens....

The ===== line will be the "center" of the conflict. All the content between the center and the <<<<<< HEAD line is content that exists in the current branch master which the HEAD ref is pointing to.

The user have to edit manually according to which lines are needed and which are not and then push the edited file again to remove the conflict.

```
$ cat merge.txt
<<<<<< HEAD
this is some content to mess with
content to append
=====
totally different content to merge later
>>>>>> new_branch_to_merge_later
```

\*Above screenshot is copied from a website since it is difficult to generate a conflict with single user.

Submitted by

Abhishek manoharan  
TVE17CS002  
Roll no. 2

