

# Introduction to Version Control

## Que 1- Git Setup

<https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html>

Ans :- Git is working

```
Please make sure you have the correct access rights
and the repository exists.
ttn@ttn:~/learnings$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status


grow, mark and tweak your common history
    branch     List, create, or delete branches
    checkout   Switch branches or restore working tree files
    commit     Record changes to the repository
    diff       Show changes between commits, commit and working tree, etc
    merge      Join two or more development histories together
    rebase     Reapply commits on top of another base tip
    tag        Create, list, delete or verify a tag object signed with GPG


collaborate (see also: git help workflows)
```

## Que2 :- Initialize a Git Repository

Ans:-

```
The Edit View Search Terminal Help
ttn@ttn:~/learnings$ git init
Reinitialized existing Git repository in /home/ttn/learnings/.git/
ttn@ttn:~/learnings$
```

Que3 :- Add files to the repository

Ans 3:-

```
ttn@ttn:~/learnings$ git add hello.txt
ttn@ttn:~/learnings$ ls
hello.txt  learning.txt
ttn@ttn:~/learnings$
```

Que4 :- Unstage 1 file

Ans 4 :-

```
@ [ttn]learnings git:(sounak) -> git reset hello.txt
Unstaged changes after reset:
D      leaning.txt
@ [ttn]learnings git:(sounak) ->
```

Que5 :- Commit the file

Ans 5:-

```
master
* sounak
@ [ttn]learnings git:(sounak) -> git add .
@ [ttn]learnings git:(sounak) -> git commit -m "File is Committed"
[sounak f4b75b4] File is Committed
1 file changed, 3 deletions(-)
delete mode 100644 leaning.txt
@ [ttn]learnings git:(sounak) ->
```

Que 6:- Add a remote

Ans 6:-

```
Please make sure you have the correct access rights
and the repository exists.
@ [ttn]learnings git:(sounak) -> git clone https://github.com/sounakguptanew/assign.git
Cloning into 'assign'...
warning: You appear to have cloned an empty repository.
@ [ttn]learnings git:(sounak) ->
```

Que 7:- Undo changes to a particular file

Ans 7:-

```
commit ddf7542441792826c1b8e7aabdfdb860df7586ae (master, hello, abc)
Author: sounak <sounak.gupta@tothenew.com>
Date: Tue Feb 5 11:43:02 2019 +0530

    vdfbgvsregehgvvehgvgedug

commit c6db9d7cee4a9f4f231882175e6b7b0d3035de2b
Author: sounak <sounak.gupta@tothenew.com>
Date: Tue Feb 5 11:39:11 2019 +0530

    sdhjgdhjpgshjdghjsd

commit 22e90518c6567a20eb7d0326713c98a83a801d87
Author: sounak <sounak.gupta@tothenew.com>
Date: Tue Feb 5 11:32:55 2019 +0530

    Learning:project I am Committing my file through git
@ [ttn]learnings git:(sounak) -> git checkout 89364e0e4fac3232693aba5e3976c2cd36
8d40a hello.txt
```

Que 8:- Push changes to Github

Ans 8:-

```
@ [ttn]learnings git:(sounak) -> git commit
On branch sounak
nothing to commit, working tree clean
@ [ttn]learnings git:(sounak) -> git branch
  abc
  hello
  master
* sounak
@ [ttn]learnings git:(sounak) -> git checkout master
Switched to branch 'master'
Your branch and 'origin/master' have diverged,
and have 3 and 24 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
@ [ttn]learnings git:(master) -> git push origin sounak
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 226 bytes | 226.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:shobhitttn/learnings.git
   89364e0..f4b75b4  sounak -> sounak
@ [ttn]learnings git:(master) ->
```

Que 9:- Clone the repository

Ans 9:-

```
Please make sure you have the correct access rights
and the repository exists.
@ [ttn]learnings git:(sounak) -> git clone https://github.com/sounakguptanew/ass
gn.git
Cloning into 'assign'...
warning: You appear to have cloned an empty repository.
@ [ttn]learnings git:(sounak) -> git loh
git: 'loh' is not a git command. See 'git --help'.
```

Que 10:- Add changes to one of the copies and pull the changes in the other.

Ans 10:-

```
commit 22e90518c6567a20eb7d0326713c98a83a801d87
Author: sounak <sounak.gupta@tothenew.com>
Date: Tue Feb 5 11:32:55 2019 +0530

    Learning:project I am Committing my file through git
@ [ttn]learnings git:(master) -> cat learning.txt
hello world git updated datsjhdgshgjdghjsjg
few changes added in file
for assignment
@ [ttn]learnings git:(master) -> git add learning.txt
@ [ttn]learnings git:(master) -> git commit
Aborting commit due to empty commit message.
@ [ttn]learnings git:(master) -> git commit -m "File Updated"
[master b932f22] File Updated
    1 file changed, 3 insertions(+), 1 deletion(-)
@ [ttn]learnings git:(master) -> git push origin sounak
Everything up-to-date
@ [ttn]learnings git:(master) -> █
```



Que 11:- Check differences between a file and its staged version

Ans 11:-

```
@ [ttn]learnings git:(master) -> ls
assign hello.txt leaning.txt learning.txt
@ [ttn]learnings git:(master) -> cat hello.txt
file creating hello.txt
@ [ttn]learnings git:(master) -> vi hello.txt
@ [ttn]learnings git:(master) -> cat hello.txt
file creating hello.txt
Some Changes added
@ [ttn]learnings git:(master) -> git add hello.txt
@ [ttn]learnings git:(master) -> git diff --staged
diff --git a/hello.txt b/hello.txt
index 981ba85..772f8a4 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,2 @@
 file creating hello.txt
+Some Changes added
@ [ttn]learnings git:(master) -> █
```

Que 12 :- Ignore a few files to be checked in

Ans 12:-

```
@ [ttn]learnings git:(master) -> touch .gitignore
@ [ttn]learnings git:(master) -> vi .gitignore
@ [ttn]learnings git:(master) -> cat .gitignore
notToBeInRepo.txt

@ [ttn]learnings git:(master) -> touch notToBeInRepo.txt
@ [ttn]learnings git:(master) -> vi notToBeInRepo.txt
@ [ttn]learnings git:(master) -> cat notToBeInRepo.txt
This will not be in repo.

@ [ttn]learnings git:(master) -> git add .gitignore
@ [ttn]learnings git:(master) -> git status
On branch master
Your branch and 'origin/master' have diverged,
and have 4 and 24 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitignore
    modified:   hello.txt

@ [ttn]learnings git:(master) -> git add notToBeInRepo.txt
The following paths are ignored by one of your .gitignore files:
notToBeInRepo.txt
Use -f if you really want to add them.
@ [ttn]learnings git:(master) -> █
```

:-

Que 13:- Create a new branch.

Ans 13:-

```
Use -f if you really want to add them.
@ [ttn]learnings git:(master) -> git checkout newbranch
error: Your local changes to the following files would be overwritten by checkout:
    hello.txt
Please commit your changes or stash them before you switch branches.
Aborting
@ [ttn]learnings git:(master) -> git commit -m "changes occur"
[master 23fd5b2] changes occur
 2 files changed, 3 insertions(+)
 create mode 100644 .gitignore
@ [ttn]learnings git:(master) -> git checkout newbranch
Branch 'newbranch' set up to track remote branch 'newbranch' from 'origin'.
Switched to a new branch 'newbranch'
@ [ttn]learnings git:(newbranch) -> █
```

Que14 :- Diverge them with commits

Ans 14:-

```
On branch sounak
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   learning.txt

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

    assign/
    notToBeInRepo.txt

no changes added to commit (use "git add" and/or "git commit -a")
@ [ttn]learnings git:(sounak) -> git add learning.txt
@ [ttn]learnings git:(sounak) -> ls
assign hello.txt learning.txt notToBeInRepo.txt
@ [ttn]learnings git:(sounak) -> git status
On branch sounak
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   learning.txt

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

    assign/
    notToBeInRepo.txt
```

Que 15:- Edit the same file at the same line on both branches and commit

Ans 15:-

```

@ [ttn]learnings git:(master|MERGING) -> git add hello.txt
@ [ttn]learnings git:(master|MERGING) -> git merge
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
@ [ttn]learnings git:(master|MERGING) -> git merge sounak
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
@ [ttn]learnings git:(master|MERGING) ->
@ [ttn]learnings git:(master|MERGING) -> git commit -m "added hellow"
[master ee24b68] added hellow
@ [ttn]learnings git:(master) -> git push origin

To github.com:shobhitttn/learnings.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:shobhitttn/learnings.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Que 16:- Try merging and resolve merge conflicts

Ans 16:-

```

and have 3 and 24 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
@ [ttn]learnings git:(master) -> git merge sounak
Auto-merging learning.txt
CONFLICT (content): Merge conflict in learning.txt
Removing leaning.txt
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.
@ [ttn]learnings git:(master|MERGING) -> █
```

Que 17:- Stash the changes and pop them

Ans 17 :-

```
@ [ttn]learnings git:(newbranch) -> git stash
Saved working directory and index state WIP on newbranch: 0579323 another changes by mudit
@ [ttn]learnings git:(newbranch) -> git commit -m "new modification updated for stash"
On branch newbranch
Your branch is up to date with 'origin/newbranch'.

Untracked files:
  notToBeInRepo.txt

nothing added to commit but untracked files present
@ [ttn]learnings git:(newbranch) -> git push origin newbranch
To github.com:shobhitttn/learnings.git
 ! [rejected]          newbranch -> newbranch (fetch first)
error: failed to push some refs to 'git@github.com:shobhitttn/learnings.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
@ [ttn]learnings git:(newbranch) -> git stash pop
On branch newbranch
Your branch is up to date with 'origin/newbranch'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   hello.txt

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

    notToBeInRepo.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (0d3066405feed7f0847d12f5ba70e76ff2027d4e)
@ [ttn]learnings git:(newbranch) -> █
```



**Que18:-** Add the following code to your .bashrc file : color\_prompt="yes"

```
parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*)/(\\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
```

**Ans:-**

```
#Git branch to be shown in bracket while in console
parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*)/(\\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\]$(parse_git_branch)\[\033[00m\]\$ '
else
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\W$(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
RED="\e[0;31m"
GREEN="\e[0;92m"
BLACK="\e[m"
YELLOW="\e[0;93m"
export PS1='\[\e[0;96m\]0 \[\e[0;94m\]\u\[\e[0;96m\]\]\W\[\e[m\]\
$(echo $(__git_ps1 "\["'$GREEN'\]' git:([\[$RED'\]\%s[\[$GREEN'\]])') \["'$YELLOW'\]' -> \["'$BLACK'\]' '

135,54 Bot
```