

## Version Control Exercises

q1) Git Setup <https://confluence.atlassian.com>.

Commands used

sudo apt install git

```
ttn@ttn: ~
File Edit View Search Terminal Help
ttn@ttn:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 14 not upgraded.
Need to get 3,907 kB of archives.
After this operation, 32.2 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.4 [3,907 kB]
Fetched 3,907 kB in 3s (1,462 kB/s)
Selecting previously unselected package git.
(Reading database ... 211023 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.4_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.4) ...
Setting up git (1:2.17.1-1ubuntu0.4) ...

Progress: [ 83%] [########################################.....]
```

```
ttn@ttn: ~/gitProject
File Edit View Search Terminal Help
ttn@ttn:~/gitProject$ git --version
git version 2.17.1
ttn@ttn:~/gitProject$ 
```

```
ttn@ttn: ~
File Edit View Search Terminal Help
ttn@ttn:~$ git config --global user.name souvikdgreat
ttn@ttn:~$ git config --global user.email souvikc40@gmail.com
ttn@ttn:~$ 
```

q2) Initialize a Git Repository

command used

git init

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git init
Initialized empty Git repository in /home/ttn/gitDir/.git/
ttn@ttn:~/gitDir$ ls -a
. . . .git
ttn@ttn:~/gitDir$ 
```

q3)Add files to the repository

commands used

//adding remote to link local repository with the remote repository

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git remote add remotel https://github.com/souvikdgreat/ttnBoot
campSouvik.git
ttn@ttn:~/gitDir$ git remote
remotel
ttn@ttn:~/gitDir$ 
```

//adding files to the remote repository

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
/home/ttn/gitDir
ttn@ttn:~/gitDir$ touch file1
ttn@ttn:~/gitDir$ ls
file1
ttn@ttn:~/gitDir$ cat >> file1
welcome to the world of distributed version control
^C
ttn@ttn:~/gitDir$ git status
On branch master

No commits yet

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

    file1

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

//adding files to the staging area

git add .

git status

git commit

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git status
On branch master

No commits yet

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

    new file:   file1

ttn@ttn:~/gitDir$ git commit -m "first commit done"
[master (root-commit) 98d2378] first commit done
 1 file changed, 1 insertion(+)
  create mode 100644 file1
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$ 
```

q4)Unstage 1 file

//edit the file in the repository and then stage the file and check status

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ ls
file1
ttn@ttn:~/gitDir$ cat >> file1
edited file1 once
^C
ttn@ttn:~/gitDir$ git status
On branch master
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:   file1

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file1

ttn@ttn:~/gitDir$
```

//unstage the file using git reset file name command

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git reset file1
Unstaged changes after reset:
M     file1
ttn@ttn:~/gitDir$ git status
On branch master
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:   file1

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$
```

q5)Commit the file

git add .

git commit -m “2nd commit done”

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "2nd commit done "
[master ff7192e] 2nd commit done
 1 file changed, 1 insertion(+)
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$
```

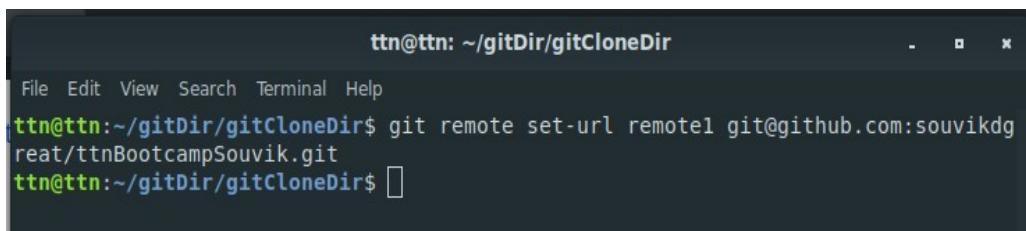
q6)Add a remote

//adding remote

git remote add remote1 <url of the repository>

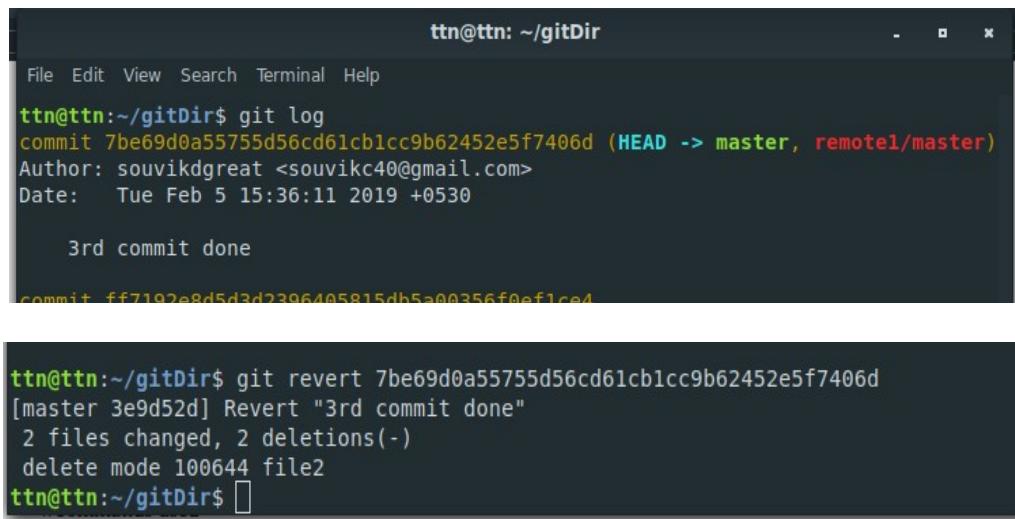
```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git remote add remote1 https://github.com/souvikdgreat/ttnBoot
campSouvik.git
ttn@ttn:~/gitDir$ git remote
remote1
ttn@ttn:~/gitDir$
```

//editing remote  
command used : git remote set-url remote <ssh url>



```
ttn@ttn: ~/gitDir/gitCloneDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir/gitCloneDir$ git remote set-url remote1 git@github.com:souvikdgreat/ttnBootcampSouvik.git
ttn@ttn:~/gitDir/gitCloneDir$ 
```

q7) Undo changes to a particular file  
//git log to display the latest commit  
//and then reverting the commit using git revert  
//and then pushing it again to the remote directory  
//commands used  
git log  
git revert commit\_id



```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git log
commit 7be69d0a55755d56cd61cb1cc9b62452e5f7406d (HEAD -> master, remote1/master)
Author: souvikdgreat <souvikc40@gmail.com>
Date:   Tue Feb 5 15:36:11 2019 +0530

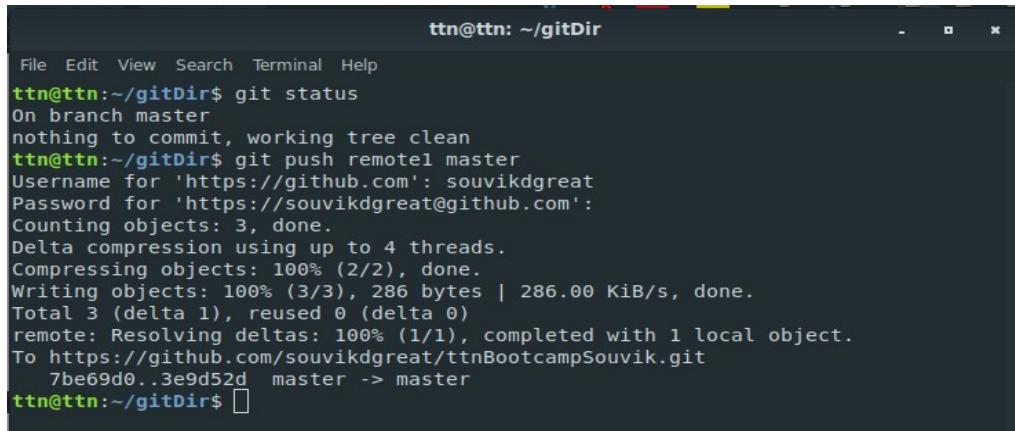
    3rd commit done

commit ff7192e8d5d3d2396405815db5a00356f0ef1ce4
```

```
ttn@ttn:~/gitDir$ git revert 7be69d0a55755d56cd61cb1cc9b62452e5f7406d
[master 3e9d52d] Revert "3rd commit done"
 2 files changed, 2 deletions(-)
 delete mode 100644 file2
ttn@ttn:~/gitDir$ 
```

q8) Push changes to Github



```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$ git push remote1 master
Username for 'https://github.com': souvikdgreat
Password for 'https://souvikdgreat@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/souvikdgreat/ttnBootcampSouvik.git
 7be69d0..3e9d52d  master -> master
ttn@ttn:~/gitDir$ 
```

## q9)Clone the repository

```
ttn@ttn: ~/gitDir/gitCloneDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir/gitCloneDir$ git clone https://github.com/souvikdgreat/ttnBootcampSouvik.git
Cloning into 'ttnBootcampSouvik'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 24 (delta 5), reused 18 (delta 3), pack-reused 0
Unpacking objects: 100% (24/24), done.
ttn@ttn:~/gitDir/gitCloneDir$
```

## q10)Add changes to one of the copies and pull the changes in the other.

//editing the files and then pushing the files to the remote server and then pulling the changes to the cloned version

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git status
On branch master
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:   file1

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

    gitCloneDir/

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$ git add .
warning: adding embedded git repository: gitCloneDir/ttnBootcampSouvik
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> gitCloneDir/ttnBootcampSouvik
hint:
hint: If you added this path by mistake, you can remove it from the
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file1
    new file:   gitCloneDir/ttnBootcampSouvik

ttn@ttn:~/gitDir$
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git commit -m "commit in order to solve question 10 "
[master 9a7f995] commit in order to solve question 10
 2 files changed, 2 insertions(+)
 create mode 160000 gitCloneDir/ttnBootcampSouvik
ttn@ttn:~/gitDir$ git push remotel master
Username for 'https://github.com': souvikdgreat
Password for 'https://souvikdgreat@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 431 bytes | 431.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/souvikdgreat/ttnBootcampSouvik.git
  3e9d52d..9a7f995  master -> master
ttn@ttn:~/gitDir$ 
```

```
ttn@ttn: ~/gitDir/gitCloneDir/ttnBootcampSouvik
File Edit View Search Terminal Help
ttn@ttn:~/gitDir/gitCloneDir/ttnBootcampSouvik$ git remote
origin
ttn@ttn:~/gitDir/gitCloneDir/ttnBootcampSouvik$ git pull origin master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
Unpacking objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
From https://github.com/souvikdgreat/ttnBootcampSouvik
 * branch            master      -> FETCH_HEAD
   3e9d52d..9a7f995  master      -> origin/master
Updating 3e9d52d..9a7f995
Fast-forward
 file1                | 1 +
 gitCloneDir/ttnBootcampSouvik | 1 +
 2 files changed, 2 insertions(+)
 create mode 160000 gitCloneDir/ttnBootcampSouvik
ttn@ttn:~/gitDir/gitCloneDir/ttnBootcampSouvik$ 
```

q11)Check differences between a file and its staged version

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
+added another line to solve question 11
ttn@ttn:~/gitDir$
ttn@ttn:~/gitDir$ git status
On branch master
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:   file1
    modified:   gitCloneDir/ttnBootcampSouvik (new commits)

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   file1
    modified:   gitCloneDir/ttnBootcampSouvik

ttn@ttn:~/gitDir$ git diff --cached file1
diff --git a/file1 b/file1
index 05f5ecc..b4ed26c 100644
--- a/file1
+++ b/file1
@@ -1,3 +1,4 @@
 welcome to the world of distributed version control
 edited file1 once
 added new line to solve question number 10
+added another line to solve question 11
```

q12)Ignore a few files to be checked in  
adding gitignore file to the repository

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ ls
file1 gitCloneDir
ttn@ttn:~/gitDir$ touch .gitignore.txt
ttn@ttn:~/gitDir$ ls
file1 gitCloneDir
ttn@ttn:~/gitDir$ ls -a
. .. file1 .git gitCloneDir .gitignore.txt
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "adding gitignore file"
[master b4f14c8] adding gitignore file
 3 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 .gitignore.txt
ttn@ttn:~/gitDir$ git push remote master
The authenticity of host 'github.com (192.30.253.112)' can't be established.
RSA key fingerprint is SHA256:Thbg6kXUpJWGl7E1IGOCspRomTxzCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.112' (RSA) to the list of known hosts.
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 429 bytes | 429.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:souvikdgreat/ttnBootcampSouvik.git
 9a7f995..b4f14c8 master -> master
ttn@ttn:~/gitDir$
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ touch a.class b.class c.log
ttn@ttn:~/gitDir$ ls
a.class b.class c.log file1 gitCloneDir
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ cat .gitignore
a.class
b.class
c.log
ttn@ttn:~/gitDir$
```

q13)Create a new branch

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git branch
* master
ttn@ttn:~/gitDir$ git checkout -b newBranch
Switched to a new branch 'newBranch'
ttn@ttn:~/gitDir$ git branch
  master
* newBranch
ttn@ttn:~/gitDir$ 
```

q14)Diverge them with commits

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git checkout master
Already on 'master'
ttn@ttn:~/gitDir$ ls
a.class  b.class  c.log  file1  gitCloneDir
ttn@ttn:~/gitDir$ touch file2
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "divergin part 1"
[master 28c6aa8] divergin part 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2
ttn@ttn:~/gitDir$ git branch
* master
  newBranch
ttn@ttn:~/gitDir$ git checkout newBranch
Switched to branch 'newBranch'
ttn@ttn:~/gitDir$ ls
a.class  b.class  c.log  file1  gitCloneDir
ttn@ttn:~/gitDir$ touch file3
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "diverging part 2"
[newBranch 8c6ee47] diverging part 2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file3
ttn@ttn:~/gitDir$ git branch
  master
* newBranch
ttn@ttn:~/gitDir$ 
```

q15)Edit the same file at the same line on both branches and commit

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git checkout master
Switched to branch 'master'
ttn@ttn:~/gitDir$ git branch
* master
  newBranch
ttn@ttn:~/gitDir$ ls
a.class  b.class  c.log  file1  file2  gitCloneDir
ttn@ttn:~/gitDir$ cat file1
welcome to the world of distributed version control
edited file1 once
added new line to solve question number 10
added another line to solve question 11
ttn@ttn:~/gitDir$ cat >>file1
adding some lines to solve merge conflicts 1
^C
ttn@ttn:~/gitDir$ cat file1
welcome to the world of distributed version control
edited file1 once
added new line to solve question number 10
added another line to solve question 11
adding some lines to solve merge conflicts 1
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "merge one commit"
[master f34329a] merge one commit
 1 file changed, 1 insertion(+)
ttn@ttn:~/gitDir$ 
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git branch
* master
newBranch
ttn@ttn:~/gitDir$ git checkout newBranch
Switched to branch 'newBranch'
ttn@ttn:~/gitDir$ ls
a.class b.class c.log file1 file3 gitCloneDir
ttn@ttn:~/gitDir$ cat file1
welcome to the world of distributed version control
edited file1 once
added new line to solve question number 10
added another line to solve question 11
ttn@ttn:~/gitDir$ cat >>file1
adding new line to check merge conflicts 2
^C
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git commit -m "merge 2 commit"
[newBranch 0c5721e] merge 2 commit
 1 file changed, 1 insertion(+)
ttn@ttn:~/gitDir$ git status
On branch newBranch
nothing to commit, working tree clean
ttn@ttn:~/gitDir$
```

q16) Try merging and resolve merge conflicts

//merging files : merging branch in master

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git branch
  master
* newBranch
ttn@ttn:~/gitDir$ git checkout master
Switched to branch 'master'
ttn@ttn:~/gitDir$ git merge newBranch
Auto-merging file1
CONFLICT (content): Merge conflict in file1
Automatic merge failed; fix conflicts and then commit the result.
ttn@ttn:~/gitDir$
```

//resolving merge conflict

```
File Edit Selection Find View Goto Tools Project Preferences Help
◀ ▶ commands ● //github commands ● file1 ×
1 welcome to the world of distributed version control
2 edited file1 once
3 added new line to solve question number 10
4 added another line to solve question 11
5 <<<<< HEAD
6 adding some lines to solve merge conflicts 1
7 =====
8 adding new line to check merge conflicts 2
9 >>>>> newBranch
10
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:

      new file:   file3

Unmerged paths:
  (use "git add <file>..." to mark resolution)

      both modified:   file1

ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

      modified:   file1
      new file:   file3

ttn@ttn:~/gitDir$ git commit -m "solved merge conflict"
[master 79e5e3c] solved merge conflict
ttn@ttn:~/gitDir$
ttn@ttn:~/gitDir$
```

q17)Stash the changes and pop them

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ ls
a.class b.class c.log file1 file2 file3  gitCloneDir
ttn@ttn:~/gitDir$ cat >>file2
this is an update to solve question no 17 of version control
^C
ttn@ttn:~/gitDir$ cat file2
this is an update to solve question no 17 of version control
ttn@ttn:~/gitDir$ git status
On branch master
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:   file2

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$
```

//after stashing

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git stash
Saved working directory and index state WIP on master: 79e5e3c solved merge conflict
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$ ls
a.class b.class c.log file1 file2 file3  gitCloneDir
ttn@ttn:~/gitDir$ cat >> file3
again updating to solve the question
17^C
ttn@ttn:~/gitDir$ cat file3
again updating to solve the question
ttn@ttn:~/gitDir$ git status
On branch master
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:   file3

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$
```

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$ git stash pop
On branch master
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:   file2

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (99f279306ca6a932a8518a3ba8906bed2cda9748)
ttn@ttn:~/gitDir$ git status
On branch master
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:   file2

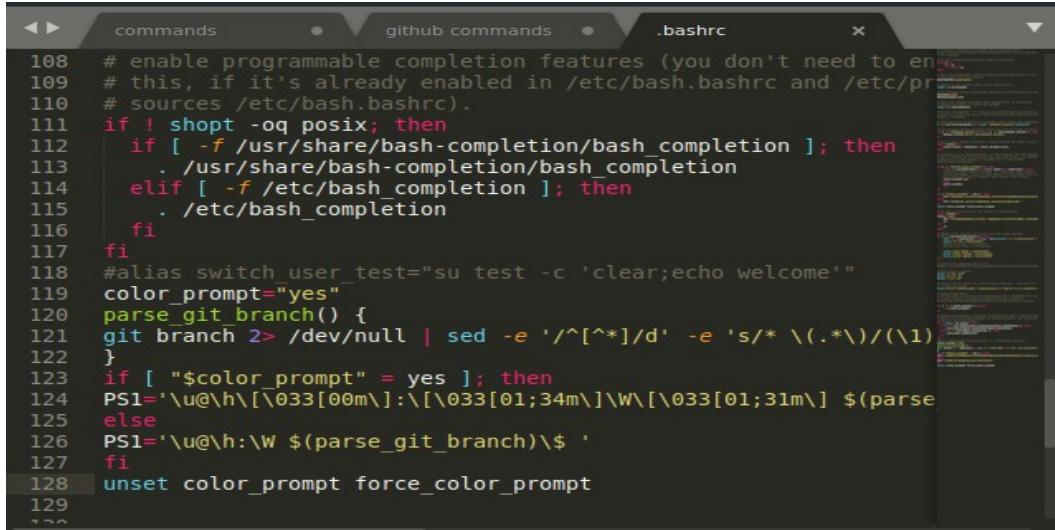
no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitDir$ git add .
ttn@ttn:~/gitDir$
```

//after git stash pop

```
ttn@ttn: ~/gitDir
File Edit View Search Terminal Help
ttn@ttn:~/gitDir$ git commit -m "commit after popping the stash"
[master 020551d] commit after popping the stash
 1 file changed, 1 insertion(+)
ttn@ttn:~/gitDir$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitDir$ git push remotel master
Warning: Permanently added the RSA host key for IP address '192.30.253.113' to t
he list of known hosts.
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 673 bytes | 673.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:souvikdgreat/ttnBootcampSouvik.git
 79e5e3c..020551d master -> master
ttn@ttn:~/gitDir$
```

q18)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[  
else  
PS1="\u@\h:\W \$(parse\_git\_branch)\$ "  
fi  
unset color\_prompt force\_color\_prompt

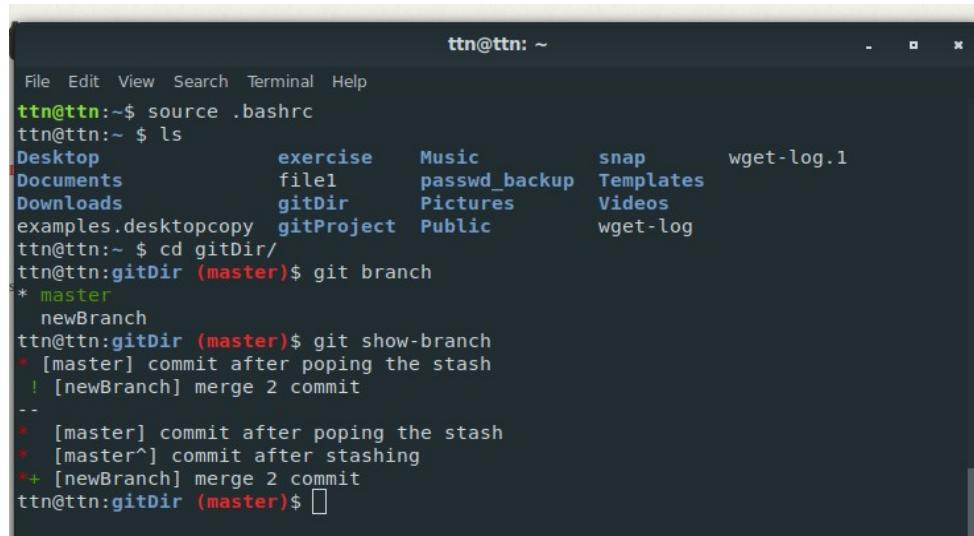
//editing the .bashrc file



A screenshot of a terminal window titled ".bashrc". The window shows a code editor with several tabs: "commands", "github commands", and ".bashrc". The ".bashrc" tab is active and displays the contents of the .bashrc file. The file contains various shell scripts and color prompts. The code editor has syntax highlighting and a dark theme.

```
108 # enable programmable completion features (you don't need to enable this, if it's already enabled in /etc/bash.bashrc and /etc/profile).
109 # sources /etc/bash.bashrc).
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118 alias switch user_test="su test -c 'clear;echo welcome'"
119 color_prompt=yes
120 parse_git_branch() {
121     git branch 2> /dev/null | sed -e '/^*/d' -e 's/* \(.*\)/(\1)'
122 }
123 if [ "$color_prompt" = yes ]; then
124     PS1='\u@\h[\033[00m]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\$ '
125 else
126     PS1='\u@\h:\w $(parse_git_branch)\$ '
127 fi
128 unset color_prompt force_color_prompt
129 ``
```

//after effect of editing is the change in color and display of the terminal



A screenshot of a terminal window titled "ttn@ttn: ~". The window shows a command-line interface with color-coded output. The user runs "source .bashrc" and then lists files in the current directory. They then change into a "gitDir" folder and run "git branch" to see two branches: "master" and "newBranch". Finally, they run "git show-branch" which displays a detailed history of commits, including stash operations and merges between the two branches. The terminal has a dark theme and includes a menu bar at the top.

```
ttn@ttn:~$ source .bashrc
ttn@ttn:~ $ ls
Desktop           exercise   Music       snap      wget-log.1
Documents        file1      passwd_backup Templates
Downloads        gitDir     Pictures    Videos
examples.desktopcopy gitProject Public      wget-log
ttn@ttn:~ $ cd gitDir/
ttn@ttn:gitDir (master)$ git branch
* master
  newBranch
ttn@ttn:gitDir (master)$ git show-branch
* [master] commit after popping the stash
! [newBranch] merge 2 commit
-- 
* [master] commit after popping the stash
* [master^] commit after stashing
*+ [newBranch] merge 2 commit
ttn@ttn:gitDir (master)$ 
```