

Version Control Exercise

Q1. Git setup.

Solution:

Git Installation:

```
ttn@ttn: ~  
File Edit View Search Terminal Help  
ttn@ttn:~$ sudo apt-get install git  
sudo: apt-get-install: command not found  
ttn@ttn:~$ sudo apt-get install git  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  git-man liberror-perl  
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 git-man liberror-perl  
0 upgraded, 3 newly installed, 0 to remove and 16 not upgraded.  
Need to get 4,733 kB of archives.  
After this operation, 33.9 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.  
17025-1 [22.8 kB]  
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all  
1:2.17.1-1ubuntu0.4 [803 kB]  
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:  
2.17.1-1ubuntu0.4 [3,907 kB]  
Fetched 4,733 kB in 3s (1,697 kB/s)
```

SSH Key Generation for GitHub Account (Without Passphrase):


```
ttn@ttn: ~  
File Edit View Search Terminal Help  
ttn@ttn:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ttn/.ssh/id_rsa): myKey  
myKey already exists.  
Overwrite (y/n)? y  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in myKey.  
Your public key has been saved in myKey.pub.  
The key fingerprint is:  
SHA256:czlmswqt7+cocNQLv/1SjmbVPyA6ZUPHpdedmV30mCY ttn@ttn  
The key's randomart image is:  
+---[RSA 2048]---+  
|         . . . . .  
|      =.+. *B|  
|    . + +Eo=+*|  
|    o + .. +o|  
|    . oS+B   .|  
|    . ..== +. |  
|    o...+.+. |  
|    .O.O= .  |  
|    .+=O.O.  |  
+---[SHA256]-----+  
ttn@ttn:~$
```

GitHub Account Setup:

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

**myKey**
9c:e9:f7:a8:08:d7:fc:9b:93:d5:ee:8f:05:01:33:b7
Added on Feb 5, 2019
Last used within the last week — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GitHub Repository:

ShubhamKumarSA / versioncontrol

Watch 0 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 releases 1 contributor

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

Q2. Initialize a Git Repository.

Solution:

```
ttn@ttn: ~/gitrepo
File Edit View Search Terminal Help
ttn@ttn:~$ cd gitrepo
ttn@ttn:~/gitrepo$ git init
Initialized empty Git repository in /home/ttn/gitrepo/.git/
ttn@ttn:~/gitrepo$
```

Q3. Add files to the repository.

Solution:

```
ttn@ttn: ~/gitrepo
File Edit View Search Terminal Help
ttn@ttn:~/gitrepo$ cat > readme.md
this is a readme file.
^Z
[2]+  Stopped                  cat > readme.md
ttn@ttn:~/gitrepo$ git add readme.md
ttn@ttn:~/gitrepo$ git status
On branch master

No commits yet

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

        new file:   readme.md

ttn@ttn:~/gitrepo$
```

Q4. Unstage 1 file.

Solution:

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

No commits yet

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

        new file:   readme.md

ttn@ttn:~/gitrepo$ git rm --cached readme.md
rm 'readme.md'
ttn@ttn:~/gitrepo$ git status
On branch master

No commits yet

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

        readme.md

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

Q5. Commit the file.

Solution:

```
ttn@ttn: ~/gitrepo
File Edit View Search Terminal Help
ttn@ttn:~/gitrepo$ git add readme.md
ttn@ttn:~/gitrepo$ git commit -m "adding readme file"
[master (root-commit) cea2f6c] adding readme file
1 file changed, 1 insertion(+)
create mode 100644 readme.md
```

Q6. Add a remote.

Solution:

```
ttn@ttn:~/gitrepo$ git remote add origin git@github.com:ShubhamKumarSA/versioncontrol.git
```

Q7. Undo changes to a particular file.

Solution:

First getting hash code of last commit then reverting git with the hash code.

```
ttn@ttn:~/gitrepo$ git log
commit cea2f6c045feeed296e52e52ac308e5822ea2ea0 (HEAD -> master, origin/master)
Author: ShubhamKumarSA <shubhamkumarsa@outlook.com>
Date: Tue Feb 5 19:46:47 2019 +0530

    adding readme file
ttn@ttn:~/gitrepo$ git revert cea2f6c045feeed296e52e52ac308e5822ea2ea0
[master 2232e1f] Revert "adding readme file"
1 file changed, 1 deletion(-)
delete mode 100644 readme.md
ttn@ttn:~/gitrepo$
```

```
ttn@ttn: ~/gitrepo
File Edit View Search Terminal Help
GNU nano 2.9.3 /home/ttn/gitrepo/.git/COMMIT_EDITMSG
Revert "adding readme file"
This reverts commit cea2f6c045feeed296e52e52ac308e5822ea2ea0.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   deleted:    readme.md
#
[ Read 11 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Q8. Push changes to Github.

Solution:

```
ttn@ttn:~/gitrepo$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 246 bytes | 246.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:ShubhamKumarSA/versioncontrol.git
 * [new branch]      master -> master
```

Q9. Clone the repository.

Solution:

```
ttn@ttn:~$ git clone git@github.com:ShubhamKumarSA/versioncontrol.git
Cloning into 'versioncontrol'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

```
ttn@ttn:~$ ls -l versioncontrol
total 4
-rw-r--r-- 1 ttn ttn 23 Feb  5 20:28 readme.md
```

Q10. Add changes to one of the copies and pull the changes in the other.

Solution:

Creating a new directory and clone git repository in it:

```
ttn@ttn:~$ mkdir versioncontrol_copy
ttn@ttn:~$ cd versioncontrol_copy
ttn@ttn:~/versioncontrol_copy$ git clone git@github.com:ShubhamKumarSA/versioncontrol.git
Cloning into 'versioncontrol'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
ttn@ttn:~/versioncontrol_copy$
```

Other git repository:

```
ttn@ttn:~$ ls -l versioncontrol
total 4
-rw-r--r-- 1 ttn ttn 23 Feb  5 20:28 readme.md
```


Modifying versioncontrol_copy readme.md file:

```
ttn@ttn:~/versioncontrol_copy$ cd versioncontrol
ttn@ttn:~/versioncontrol_copy/versioncontrol$ ls
readme.md
ttn@ttn:~/versioncontrol_copy/versioncontrol$ echo "Added New Line" >> readme.m
d
ttn@ttn:~/versioncontrol_copy/versioncontrol$ cat readme.md
this is a readme file.
Added New Line
ttn@ttn:~/versioncontrol_copy/versioncontrol$ git add readme.md
ttn@ttn:~/versioncontrol_copy/versioncontrol$ git commit -m "Modified ReadMe Fro
m Other Repo"
[master 73d06a9] Modified ReadMe From Other Repo
 1 file changed, 1 insertion(+)
ttn@ttn:~/versioncontrol_copy/versioncontrol$
```

Now pulling the changes in other repository:

```
ttn@ttn:~/versioncontrol_copy/versioncontrol$ cd ~/versioncontrol
ttn@ttn:~/versioncontrol$ cat readme.md
this is a readme file.
ttn@ttn:~/versioncontrol$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:ShubhamKumarSA/versioncontrol
 * branch                master      -> FETCH_HEAD
   cea2f6c..73d06a9      master      -> origin/master
Updating cea2f6c..73d06a9
Fast-forward
 readme.md | 1 +
 1 file changed, 1 insertion(+)
ttn@ttn:~/versioncontrol$ cat readme.md
this is a readme file.
Added New Line
ttn@ttn:~/versioncontrol$
```

Q11. Check differences between a file and its staged version.

Solution:

Firstly made changes in readme.md file and staged it using git add command:

```
ttn@ttn:~/versioncontrol$ echo "OTHER CHNAGES" >> readme.md
ttn@ttn:~/versioncontrol$ cat readme.md
this is a readme file.
Added New Line
OTHER CHNAGES
ttn@ttn:~/versioncontrol$ git add readme.md
ttn@ttn:~/versioncontrol$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

        modified:   readme.md

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

        readme.me

ttn@ttn:~/versioncontrol$ git diff --staged
diff --git a/readme.md b/readme.md
index d0cd8b9..23aa355 100644
--- a/readme.md
+++ b/readme.md
@@ -1,2 +1,3 @@
 this is a readme file.
 Added New Line
+OTHER CHNAGES
```

Now making some more modification in readme.md and checking difference.

```
ttn@ttn:~/versioncontrol$ echo "Changes After DIFF CHECK." >> readme.md
ttn@ttn:~/versioncontrol$ git diff
diff --git a/readme.md b/readme.md
index 23aa355..e142245 100644
--- a/readme.md
+++ b/readme.md
@@ -1,3 +1,4 @@
 this is a readme file.
 Added New Line
 OTHER CHNAGES
+Changes After DIFF CHECK.
```

Q12. Ignore a few files to be checked in.

Solution:

Adding .gitignore file into git and adding a filename that is to be ignored:

```
ttn@ttn:~/versioncontrol$ ls
datafile.txt  readme.md
ttn@ttn:~/versioncontrol$ echo "datafile.txt" >> .gitignore
ttn@ttn:~/versioncontrol$ ls
datafile.txt  readme.md
ttn@ttn:~/versioncontrol$ ls -a
.  ..  datafile.txt  .git  .gitignore  readme.md
```

Now checking git status:

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

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

        modified:   readme.md

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:   readme.md

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

        .gitignore
```

Q13. Create a new branch.

Solution:

```
ttn@ttn:~/versioncontrol$ git checkout -b testingBranch
A       .gitignore
M       readme.md
Switched to a new branch 'testingBranch'
```


Q14. Diverge them with commits.

Solution:

```
ttn@ttn:~/versioncontrol$ git status
On branch testingBranch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitignore
        modified:   readme.md

ttn@ttn:~/versioncontrol$ echo "changing readme.md file for testingBranch" >> readme.md
ttn@ttn:~/versioncontrol$ git add readme.md
ttn@ttn:~/versioncontrol$ git commit -m "Diverge"
[testingBranch c3fb219] Diverge
 2 files changed, 5 insertions(+)
 create mode 100644 .gitignore
```

```
ttn@ttn:~/versioncontrol$ git push origin testingBranch
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 419 bytes | 419.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To github.com:ShubhamKumarSA/versioncontrol.git
   73d06a9..c3fb219  testingBranch -> testingBranch
ttn@ttn:~/versioncontrol$
```

Q15. Edit the same file at the same line on both branches and commit.

Solution:

testingBranch branch file 'testfile.txt' commit:

```
ttn@ttn:~/versioncontrol$ echo "Edit on Same Line testingBranch." >> testfile.txt
ttn@ttn:~/versioncontrol$ git status
On branch testingBranch
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:   readme.md

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

        testfile.txt

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

        modified:   readme.md
        new file:   testfile.txt

ttn@ttn:~/versioncontrol$ git commit -m "SAME-LINE-TB"
[testingBranch a4b758b] SAME-LINE-TB
 2 files changed, 2 insertions(+)
 create mode 100644 testfile.txt
```

File Content:

```
ttn@ttn:~/versioncontrol$ cat testfile.txt
Edit on Same Line testingBranch.
```

master repository file 'testfile.txt' commit:

```
ttn@ttn:~/versioncontrol$ echo "Edit on Same Line master." >> testfile.txt
ttn@ttn:~/versioncontrol$ git add testfile.txt
ttn@ttn:~/versioncontrol$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

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

        new file:   testfile.txt

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

        datafile.txt
        testFile.txt

ttn@ttn:~/versioncontrol$ git commit -m "SAME-LINE-MASTER"
[master a15afd5] SAME-LINE-MASTER
 1 file changed, 1 insertion(+)
 create mode 100644 testfile.txt
```

File Content:

```
ttn@ttn:~/versioncontrol$ cat testfile.txt
Edit on Same Line master.
```

Q16. Try merging and resolve merge conflicts.

Solution:

Trying to merge repositories:

```
ttn@ttn:~/versioncontrol$ git merge testingBranch
Auto-merging testfile.txt
CONFLICT (add/add): Merge conflict in testfile.txt
Auto-merging readme.md
CONFLICT (content): Merge conflict in readme.md
Automatic merge failed; fix conflicts and then commit the result.
```

Now resolving the conflicts

Git Status:

```
ttn@ttn:~/versioncontrol$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
        new file:   .gitignore

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

        both modified:   readme.md
        both added:      testfile.txt

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

        testFile.txt
```

Resolving by staging the master repository and committing it, then we try to merge the repository:

```

ttn@ttn:~/versioncontrol$ git add .
ttn@ttn:~/versioncontrol$ git commit -m "MERGE-CON-RESOLVE"
[master dda4502] MERGE-CON-RESOLVE
ttn@ttn:~/versioncontrol$ git merge
Already up to date.
ttn@ttn:~/versioncontrol$ git push origin master
Counting objects: 14, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 1.53 KiB | 785.00 KiB/s, done.
Total 14 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:ShubhamKumarSA/versioncontrol.git
73d06a9..dda4502 master -> master

```

Q17. Stash the changes and pop them.

Solution:

Adding a file stash.txt in master repository and then stashing changes and again try to pop stash in branch repository:

```

ttn@ttn: ~/versioncontrol
File Edit View Search Terminal Help
ttn@ttn:~/versioncontrol$ touch stash.html
ttn@ttn:~/versioncontrol$ git add stash.html
ttn@ttn:~/versioncontrol$ git stash
Saved working directory and index state WIP on master: f310ca9 STASH CHANGE
ttn@ttn:~/versioncontrol$ git checkout testingBranch
Switched to branch 'testingBranch'
ttn@ttn:~/versioncontrol$ touch stash.html
ttn@ttn:~/versioncontrol$ git add stash.html
ttn@ttn:~/versioncontrol$ git status
On branch testingBranch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   stash.html

ttn@ttn:~/versioncontrol$ git stash pop
On branch testingBranch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   stash.html

Dropped refs/stash@{0} (17f73e07d4137804cfd0740d5fc3bd6f305bb75c)
ttn@ttn:~/versioncontrol$ git commit -m "Stash changed"
[testingBranch 369c327] Stash changed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 stash.html
ttn@ttn:~/versioncontrol$

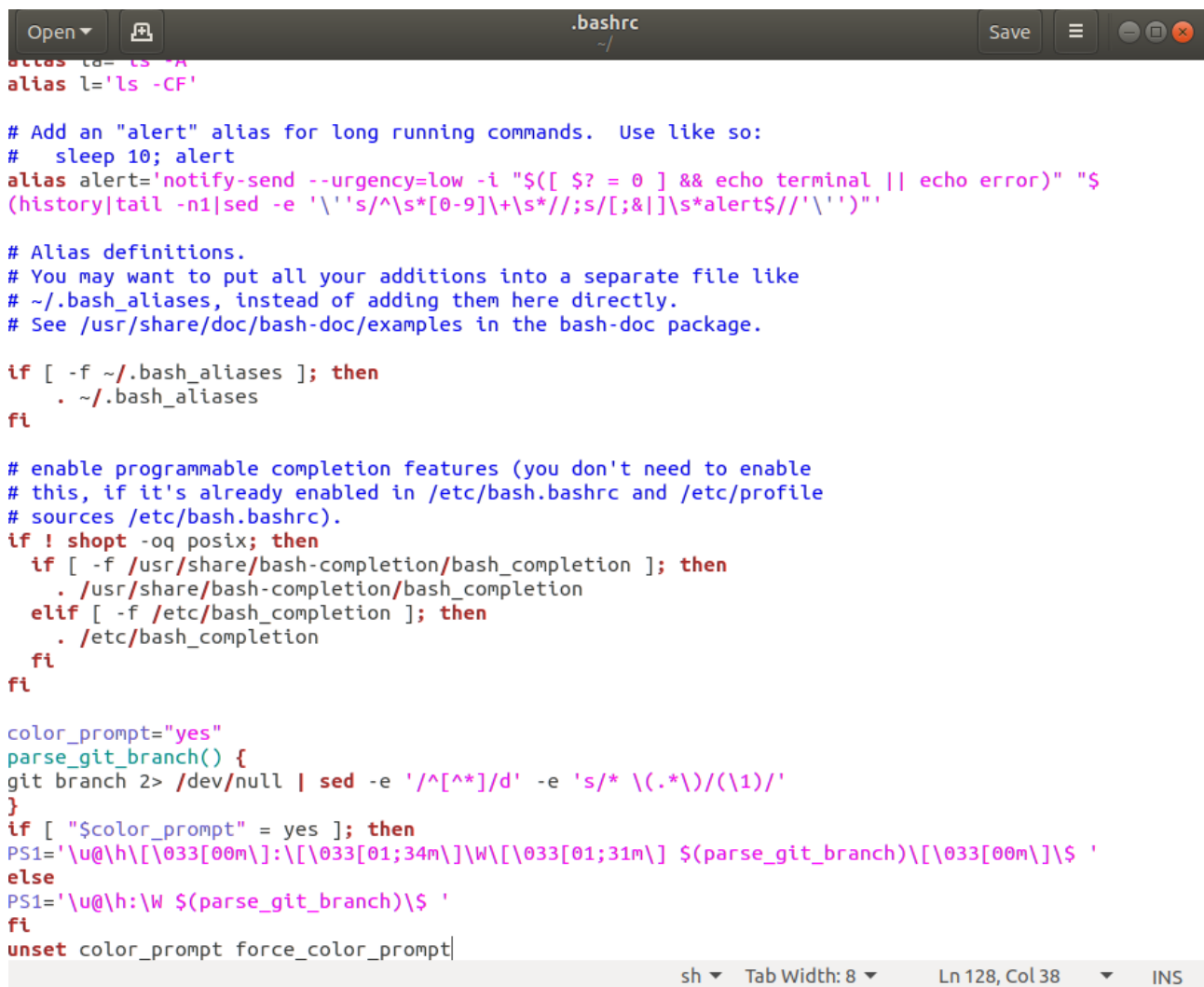
```

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\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\033[00m\]\$ '  
else  
PS1='\u@\h:\W $(parse_git_branch)\$ '  
fi  
unset color_prompt force_color_prompt
```

Solution:

Edited bashrc file.



```
.bashrc  
~/  
alias l='ls -CF'  
  
# Add an "alert" alias for long running commands. Use like so:  
# sleep 10; alert  
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$  
(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[\;|\&|]\s*alert$/'\`'\`"'')'  
  
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
    if [ -f /usr/share/bash-completion/bash_completion ]; then  
        . /usr/share/bash-completion/bash_completion  
    elif [ -f /etc/bash_completion ]; then  
        . /etc/bash_completion  
    fi  
fi  
  
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
```

```
ttn@ttn:~/versioncontrol$ gedit ~/.bashrc  
ttn@ttn:~/versioncontrol$ source ~/.bashrc  
ttn@ttn:versioncontrol (testingBranch)$
```