

Assignment 2
By: Anindya Guha

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

```
ttn@anindya:~$ sudo apt-get install git
[sudo] password for ttn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
The following packages were automatically installed and are no longer required:
  linux-image-4.15.0-29-generic linux-modules-4.15.0-29-generic
  linux-modules-extra-4.15.0-29-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
ttn@anindya:~$
```

2. Initialize a Git Repository

```
ttn@anindya:~/assignment2$ git init
Initialized empty Git repository in /home/ttn/assignment2/.git/
```

3. Add files to the repository
4. Unstage 1 file

```
ttn@anindya:~/assignment2$ touch file1
ttn@anindya:~/assignment2$ 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)
ttn@anindya:~/assignment2$
```

5.Commit the file

```
ttn@anindya:~/assignment2$ git commit -m "File added"
On branch master

Initial commit

Untracked files:
  (use "git add" to add)
        file1

nothing added to commit but untracked files present
ttn@anindya:~/assignment2$ git remote add
```

6.Add a remote

```
ttn@anindya:~/assignment2$ git remote add origin https://github.com/Ricky2494/repos.git
ttn@anindya:~/assignment2$ git remote -v
origin https://github.com/Ricky2494/repos.git (fetch)
origin https://github.com/Ricky2494/repos.git (push)
ttn@anindya:~/assignment2$
```

7.Undo changes to a particular file

```
ttn@anindya:~/assignment2$ vi file2
ttn@anindya:~/assignment2$ cat file2
hshhshs
C
C
gsgs
ttn@anindya:~/assignment2$ git checkout file2
ttn@anindya:~/assignment2$ cat file2
ffghss
hshhshs
C
C
gsgs
ttn@anindya:~/assignment2$
```

8.Push changes to Github

```

ttn@anindya:~/assignment2$ git push -u origin master
Username for 'https://github.com': Ricky2494
Password for 'https://Ricky2494@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 236 bytes | 236.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Ricky2494/repos.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ttn@anindya:~/assignment2$

```

9.Clone the repository

```

ttn@anindya:~/assignment2$ cd ~
ttn@anindya:~$ mkdir assign2
ttn@anindya:~$ cd assign2
ttn@anindya:~/assign2$ git clone https://github.com/Ricky2494/repos.git
Cloning into 'repos'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
ttn@anindya:~/assign2$

```

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

```

File Edit View Search Terminal Help
ttn@anindya:~/assign2$ cd repos
ttn@anindya:~/assign2/repos$ vi file2
ttn@anindya:~/assign2/repos$ git add file2
ttn@anindya:~/assign2/repos$ git commit -m "Changes done"
[master be6c593] Changes done
1 file changed, 1 insertion(+), 1 deletion(-)
ttn@anindya:~/assign2/repos$ git push -u origin master
Username for 'https://github.com': Ricky2494
Password for 'https://Ricky2494@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 261 bytes | 261.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Ricky2494/repos.git
 556780d..be6c593  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ttn@anindya:~/assign2/repos$ cd assignment2
bash: cd: assignment2: No such file or directory
ttn@anindya:~/assign2/repos$ cd ~
ttn@anindya:~$ cd assignment2
ttn@anindya:~/assignment2$ 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 https://github.com/Ricky2494/repos
 * branch      master      -> FETCH_HEAD
 556780d..be6c593  master    -> origin/master
Updating 556780d..be6c593
Fast-forward
 file2 | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@anindya:~/assignment2$ cat file2
hshhshs
c
c
gsgs
ttn@anindya:~/assignment2$

```

11. Check differences between a file and its staged version

```
ttn@anindya:~/assignment2$ git diff HEAD^
diff --git a/file2 b/file2
index e0fbc61..22c4f02 100644
--- a/file2
+++ b/file2
@@ -1,4 +1,4 @@
-ffghss
+
hshhshs
C
C
ttn@anindya:~/assignment2$
```

12. Ignore a few files to be checked in

```
ttn@anindya:~/assignment2$ vi .gitignore
ttn@anindya:~/assignment2$ 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:   abc

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

        .gitignore
        abc.txt
        abc1
        abc2
        abc3
        file1

ttn@anindya:~/assignment2$
```

13. Create a new branch.


```

ttn@anindya:~/assignment2$ git branch branch2
ttn@anindya:~/assignment2$ git branch
  branch2
* master
ttn@anindya:~/assignment2$ 

```

14. Diverge them with commits

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

16. Try merging and resolve merge conflicts

17. Stash the changes and pop them

```

ttn@anindya:~/assignment2$ git status
On branch branch1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        abc.txt
        abc1
        abc2
        abc3
        fil
        file1

nothing added to commit but untracked files present (use "git add" to track)
ttn@anindya:~/assignment2$ git stash
No local changes to save
ttn@anindya:~/assignment2$ git status
On branch branch1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        abc.txt
        abc1

```

```

nothing added to commit but untracked files present (use "git add" to track)
ttn@anindya:~/assignment2$ git stash -u
Saved working directory and index state WIP on branch1: 10ccd77 adding a change
from the feature branch
ttn@anindya:~/assignment2$ git status
On branch branch1
nothing to commit, working tree clean
ttn@anindya:~/assignment2$ git stash pop
Already up to date!
On branch branch1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        abc.txt
        abc1
        abc2
        abc3
        fil
        file1

nothing added to commit but untracked files present (use "git add" to track)
Dropped refs/stash@{0} (c4ec886723cf721e22b65f7377a687b7d2577e8a)
ttn@anindya:~/assignment2$ 

```

18. 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

```

```

ttn@anindya:~/assignment2$ color_prompt="yes"
ttn@anindya:~/assignment2$ parse_git_branch() {
> git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*)/(\\1)/'
> }
ttn@anindya:~/assignment2$ 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 $(parse_git_branch)\$ '
> fi
ttn@anindya:assignment2 (master)$ unset color_prompt force_color_prompt
ttn@anindya:assignment2 (master)$ 

```