

1. Difference Between Two copies of code:

```
diff file1 file2
```

Returns only the difference between the two files

< old line

> new line

```
Diff -u file1 file 2
```

+ And - symbols

2. To see changes made into files:

```
Cat file.diff
```

To apply changes

```
Patch file.py < file.diff
```

3. Fixing file

```
diff -u file1 file2 > file.diff
```

4. Git config file

```
Git config --global user.email
```

```
Git config --global user.name
```

```
Cd dir
```

```
Git init
```

```
Git add file
```

```
Git status
```

File status

Untracked

Modifiedchange

Staged.....git add

Committed....git commit

5. See git logging

```
Git log
```

6. To skip staging area

```
Git commit -a -m "....." #works only with traced files
```

7. Git log -p #changes

8. Git show commit_hash #changes

9. Git log --stat

10. Git diff -staged

11. Git rm file #delete the file
12. Git mv file1 file2 #move/rename
13. Echo .exe > .gitignore #ignore the sys generated files
14. Git checkout #undoing commit
15. Git add * to add all the files
16. To unstage something git reset HEAD file
17. Git commit --amend #amending commit msg
18. Git revert HEAD #new commit with inverse change
19. Check by git log -p -2
20. Git revert commitid

Branch:

#create branch

1. Git branch #show branches
2. Git branch name #new branches
3. Git checkout name #switched to new branch
4. Git checkout -b name2 #create and switched
5. Git branch -d branch_name #delete

#merge branch

1. Confirm u are in master
2. Git merge branch_name #uses fast-forward(linear) and three way merge(diverge) to perform merge
3. Git log to confirm
4. Git log --graph --oneline

Working In Remote

Github,bitbucket,gitlab

Git remote -v

Git remote show origin

Git fetch #download contents

Git status

Git merge origin/master

Git checkout remote_branch

#working with pull-merge-push

Git log --graph --oneline --all

Git log -p origin/master

##merging

Git checkout master

Git merge branch

Git push --delete origin branch

Git rebase

