

#After everything is set, the first thing we have to do is to configure git with our name and email:

```
git config --global user.name git:
cd desktop
mkdir mygitrepo
cd mygitrepo
```

Now we're ready to initialize a brand new git repository.

```
git init
```

#We can check for the current status of the git repository by using

```
git status
```

#We can check if there is any file in the directory by

```
ls
```

#Git directory is hidden to see it we use

```
ls -a
```

#to go to previous directory we use

```
cd ..
```

#Create and commit a new file

```
touch hello.txt
```

#To "register" the file for committing we need to add it to git using

```
git add hello.txt
```

#Checking for the status now indicates that the file is ready to be committed:

```
git status
```

#We can now commit it to the repository
git commit -m "Add my first file"

#To see recent changes/commit we use
git log

#this would add Hello, world! to hello.txt
echo Hello, world! > hello.txt

#A patch-style view of the difference between the currently edited and
committed files
git diff

#The full list of changes since the beginning of time:
git log
git log --since=yesterday
git log --since=2weeks

#To discover why, when and by whom a certain line was added
git blame hello.txt

#To abort current uncommitted changes and restore the working copy to
the last committed state:

Discards all of currently uncommitted (unstaged or staged) changes:
git reset --hard

#To remove a file from staging area we use

git reset HEAD ~filename~

#To view the statistics and facts about the last commit:

git show

#A remote called origin is automatically created if we cloned a remote repository. The full address of that remote can be viewed with:

git remote -v

#to add origin and check whether remote is added or not

git remote add origin repo link

git remote

#To put changes from local repo in the remote repo

git push origin master

#From remote repo to get most recent changes.

git pull