

Git 2

telling who is working (this is one time process)

```
akshu20791gmail@ip-172-31-83-7: ~/Desktop/project
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --global user.name akshat
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

Also we can config email Id

```
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --global user.email "akshu20791@gmail.com"
```

↓
configuring mail id with git

now if you want what is configured

```
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --list
user.name=akshat
user.email=akshu20791@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

lets play around git to understand the concepts more...lets create one file akshat.txt

```
akshu20791gmail@ip-172-31-83-7: ~/Desktop/project
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git status
On branch master

No commits yet

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

akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ touch akshat.txt
```

← create a blank file
named
akshat.txt

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/projects$ ls
akshat.txt  index.html
```

you will see that akshat.txt is created in working directory ..and if I do git status it would be in red color

```
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/projects$ git status
On branch master

No commits yet

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

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

akshu20791gmail@ip-172-31-83-7:~/Desktop/projects$ git add akshat.txt
```

or you can also use
git add .
↑
dot

Lets see what happened now if we do git status

```
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   akshat.txt
        new file:   index.html

akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

Now we lets commit the files ...which means create the version of the file...keep in your mind that all the files in staging area would be commit not just a single file

File Edit View Search Terminal Help

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git commit -m "first commit"
```

Commit the files
in staging area

message

The message we would
like to write while
creating commit

↓
Create the version (local version) of
the files present in
staging area

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git commit -m "first commit"
```

```
[master (root-commit) b686c43] first commit
```

```
2 files changed, 1 insertion(+)
```

```
create mode 100644 akshat.txt
```

```
create mode 100644 index.html
```

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

git status

```
create mode 100644 index.html
```

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

to see the versions which we have created ...

```
akshu20791gmail@ip-172-31-83-7: ~/Desktop/project
File Edit View Search Terminal Help
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git log
commit b686c435cf456f038d562c2ff097a5faed344c1a (HEAD -> master)
Author: akshat <akshu20791@gmail.com>
Date: Sat May 27 14:59:17 2023 +0000
first commit
akshu20791gmail@ip-172-31-83-7:~/Desktop/projects$
```

Shows all the commits which is done

unique commit id

*who is working
akshat with mail id
akshu2079
(we will already it)*

Message

If I have 100 commits its very difficult to see such a huge dataso I want to see the commits in single line

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git log --oneline
b686c43 (HEAD -> master) first commit
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

Alias -> giving short name to big command

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git status
On branch master
nothing to commit, working tree clean
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --global alias.st status
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git st
On branch master
nothing to commit, working tree clean
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git status
On branch master
nothing to commit, working tree clean
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

*I am setting up the
alias of the word status
as st*

Now if I use git st it would be same as git status

if you want to unset the alias

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --global --unset alias.st
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git st
git: 'st' is not a git command. See 'git --help'.
```

```
The most similar commands are
  status
  reset
  stage
  stash
```

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

Checking if
st is still alias

Removing the st as alias
for status

lets say I want to set alias of commit -m in git commit -m "message" as git cm "message"

```
File Edit View Search Terminal Help
```

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git config --global alias.cm 'commit -m'
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ touch file1
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git add
```

```
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$ git cm "second commit"
[master be314c6] second commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1
akshu20791gmail@ip-172-31-83-7:~/Desktop/project$
```

Setting up alias
a file in repo

adding that file in staging area

committing and creating a new version