'GIT'

Cheat sheet

Git is a version control system that tracks changes in every file taking place in your computer locally

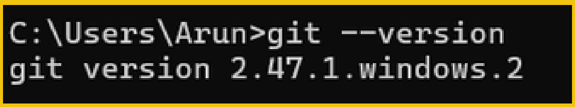
Github is a website that allows developers to upload,store and manage their projects

‘GIT’

Cheat sheet

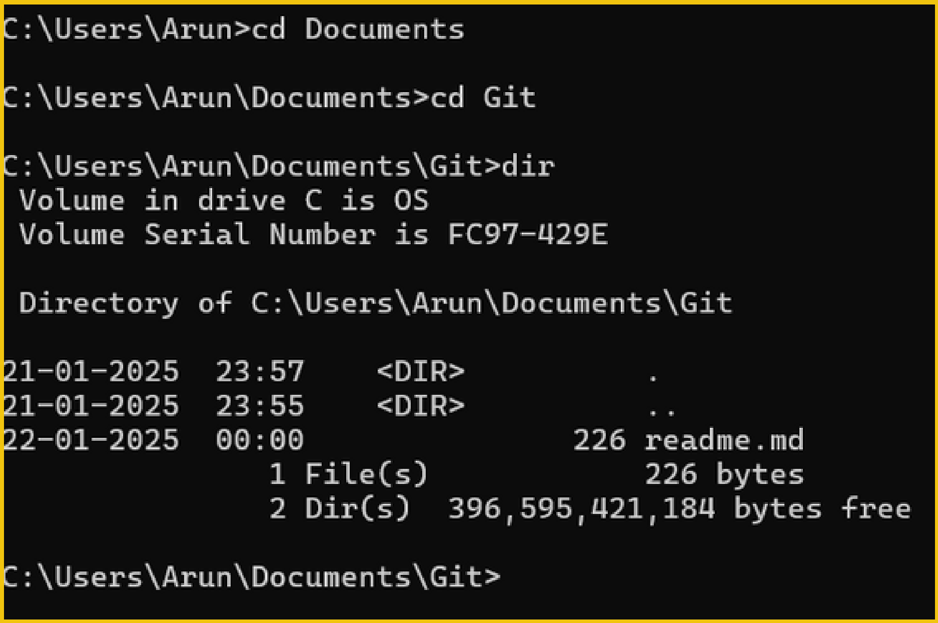
Step 1 : download git

Step 2 : check version



```
C:\Users\Arun>git --version
git version 2.47.1.windows.2
```

Step 3: Getting into specific directory and view files



```
C:\Users\Arun>cd Documents

C:\Users\Arun\Documents>cd Git

C:\Users\Arun\Documents\Git>dir
 Volume in drive C is OS
 Volume Serial Number is FC97-429E

 Directory of C:\Users\Arun\Documents\Git

21-01-2025  23:57    <DIR>          .
21-01-2025  23:55    <DIR>          ..
22-01-2025  00:00               226 readme.md
               1 File(s)            226 bytes
               2 Dir(s)  396,595,421,184 bytes free

C:\Users\Arun\Documents\Git>
```

Basic commands :

‘*Cls*’ : clear the terminal

‘*cd..*’ : get out of current directory

‘*ls*’ : get list of all files in current directory

‘*ls -a*’ : get list of all files + hidden files (in git bash)

‘*ls -Force*’ : get list of all files + hidden files (in powershell)

‘GIT’

Cheat sheet

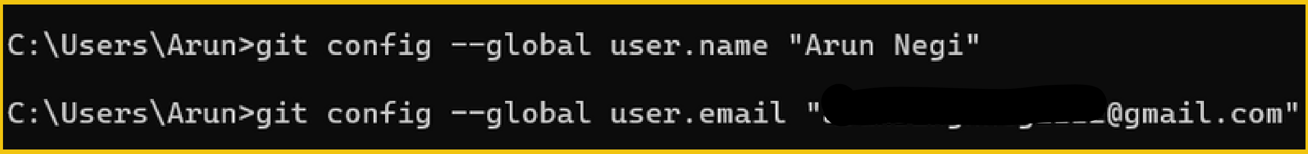
Configuring Git :

setting up your username and email, and defining behaviors like default editors or merge tools.

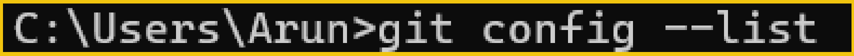
Two types of configuration :

Global: Configurations that apply to all repositories on your computer (e.g., username/email).

Local: Configurations specific to a single repository (e.g., project-specific username/email).



```
C:\Users\Arun>git config --global user.name "Arun Negi"

C:\Users\Arun>git config --global user.email "[redacted]@gmail.com"
```



```
C:\Users\Arun>git config --list
```

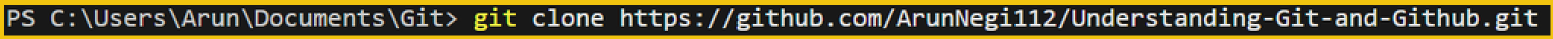
-- To get configuration details

Clone :

Clones the whole remote repo to local system

command :

git clone paste_link_here (Note: Do this in VS studio terminal)



```
PS C:\Users\Arun\Documents\Git> git clone https://github.com/ArunNegi112/Understanding-Git-and-Github.git
```

‘GIT’

Cheat sheet

Status :

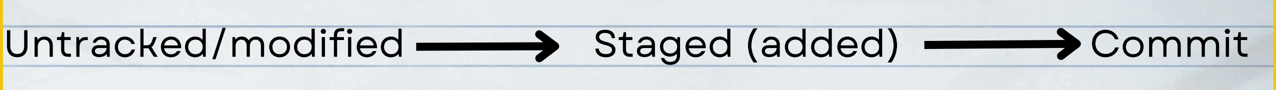
Four types of status :

Modified : when we modify a file but doesn’t commit to git

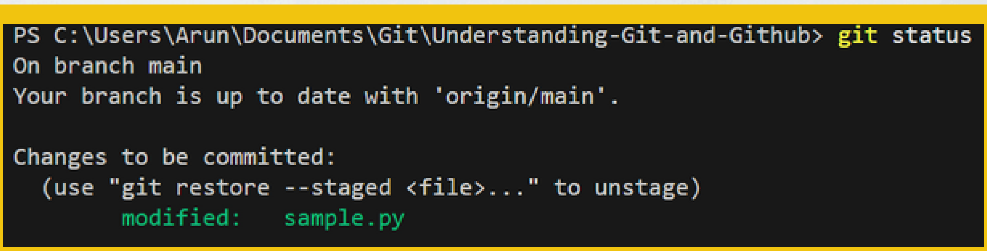
Untracked : creating new file, that is not yet committed

Staged : file is ready to commit (i.e we’ve added the file)

Unmodified : file is unchanged or the change has been committed



Untracked/modified → Staged (added) → Commit

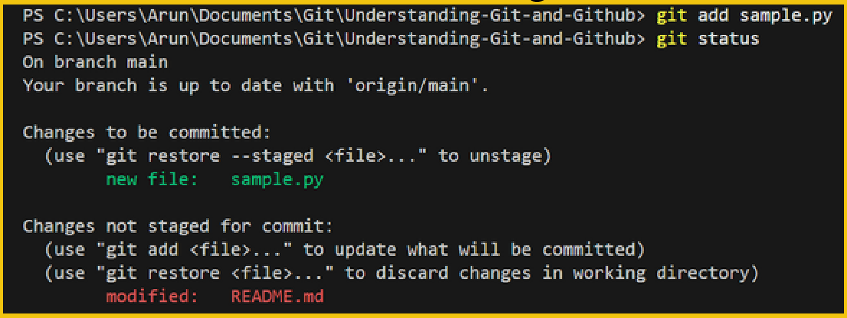


```
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   sample.py
```

Add, Commit and push :

‘*git add file_name*’

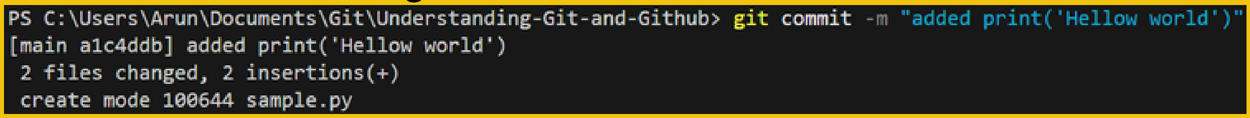
if want to add all files : ‘*git add .*’



```
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github> git add sample.py
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sample.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
```

‘*git commit -m “commit message”*

Will commit all staged files



```
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github> git commit -m "added print('Hellow world')"
[main a1c4ddb] added print('Hellow world')
 2 files changed, 2 insertions(+)
 create mode 100644 sample.py
```

Upload local repo content to remote repo (github)

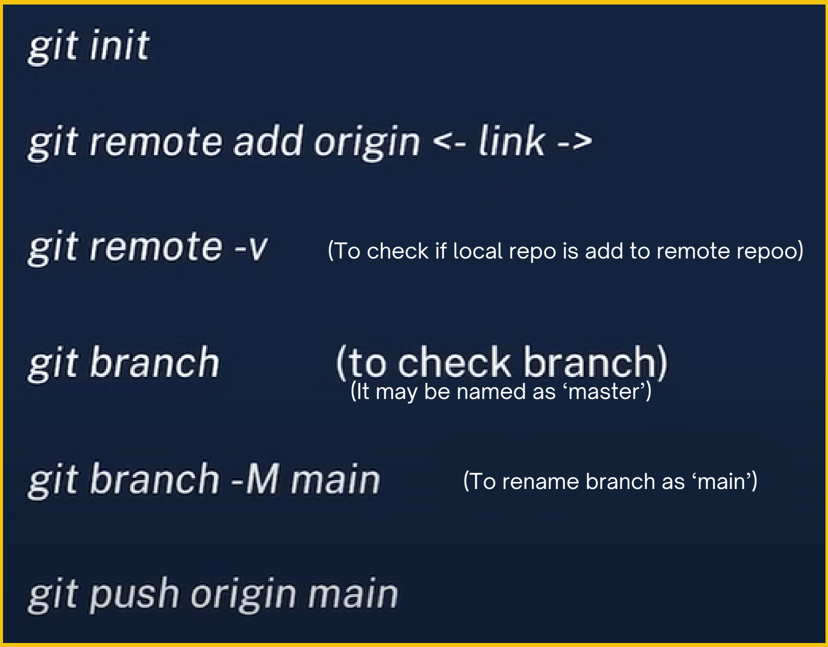
‘*git push origin main*’ : ‘origin’ is default name of that remote repo(can be changed later), ‘main’ is the branch we are pushing into (can be different

'GIT'

Cheat sheet

Uploading local *repo* to remote repo:

Create a new repository on github ---> copy its http link ---> follow the steps below in vs code terminal

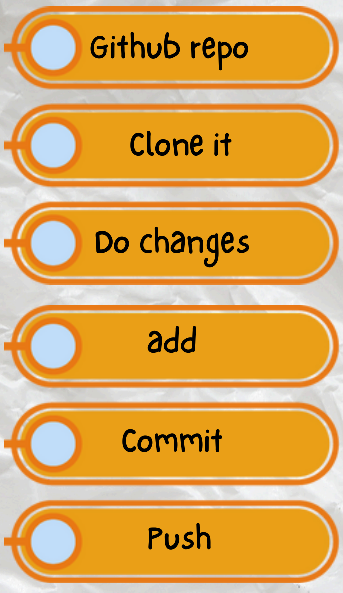


```
git init

git remote add origin <- link ->

git remote -v          (To check if local repo is add to remote repoo)

git branch             (to check branch)
                        (It may be named as 'master')

git branch -M main          (To rename branch as 'main')

git push origin main
```

‘GIT’

Cheat sheet

GIT WorkFlow



Github repo

Clone it

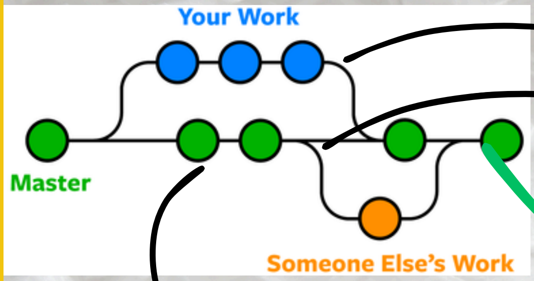
Do changes

add

Commit

Push

Branches



lets say this branch is me working on feature 1

this branch is someone else doing his work on feature 2

This is ‘merge’, after completion of all individual work, we merge the codes to complete one single project

This is the main/master branch

Commands

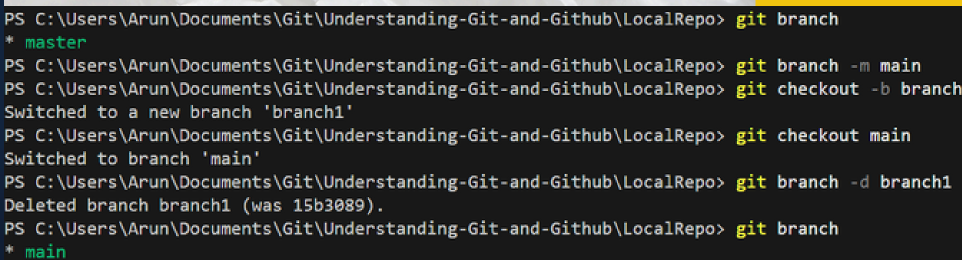
git branch (to check branch)

git branch -M main (to rename branch)

git checkout <- branch name -> (to navigate)

git checkout -b <- new branch name -> (to create new branch)

git branch -d <- branch name -> (to delete branch)



```
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git branch
* master
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git branch -m main
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git checkout -b branch
Switched to a new branch 'branch1'
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git checkout main
Switched to branch 'main'
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git branch -d branch1
Deleted branch branch1 (was 15b3089).
PS C:\Users\Arun\Documents\Git\Understanding-Git-and-Github\LocalRepo> git branch
* main
```