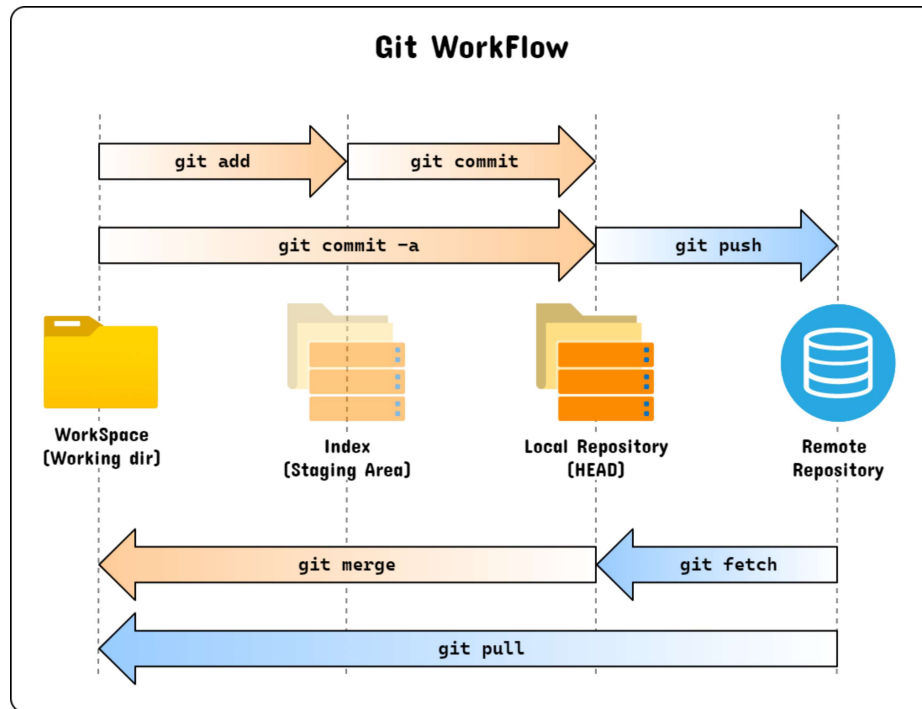


**Experiment No : 03**

**Aim : To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet**

**Theory :**

**SETUP & INIT**

Configuring user information, initializing and cloning repositories

❖ **git init**

initialize an existing directory as a Git repository.

```

Windows PowerShell
PS C:\Users\vaibh\Desktop> mkdir DemoRepo

Directory: C:\Users\vaibh\Desktop

Mode                LastWriteTime         Length Name
----                -
d-----          10-08-2021  11:18 AM             DemoRepo

PS C:\Users\vaibh\Desktop> cd DemoRepo
PS C:\Users\vaibh\Desktop\DemoRepo> git init
Initialized empty Git repository in C:/Users/vaibh/Desktop/DemoRepo/.git/
PS C:\Users\vaibh\Desktop\DemoRepo>

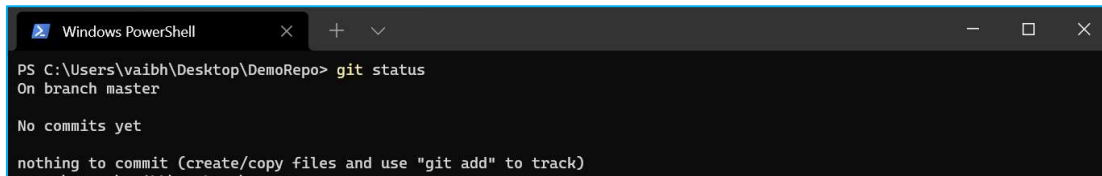
```

## STAGE & SNAPSHOT

Working with snapshots and the Git staging area

### ❖ git status

show modified files in working directory, staged for your next commit



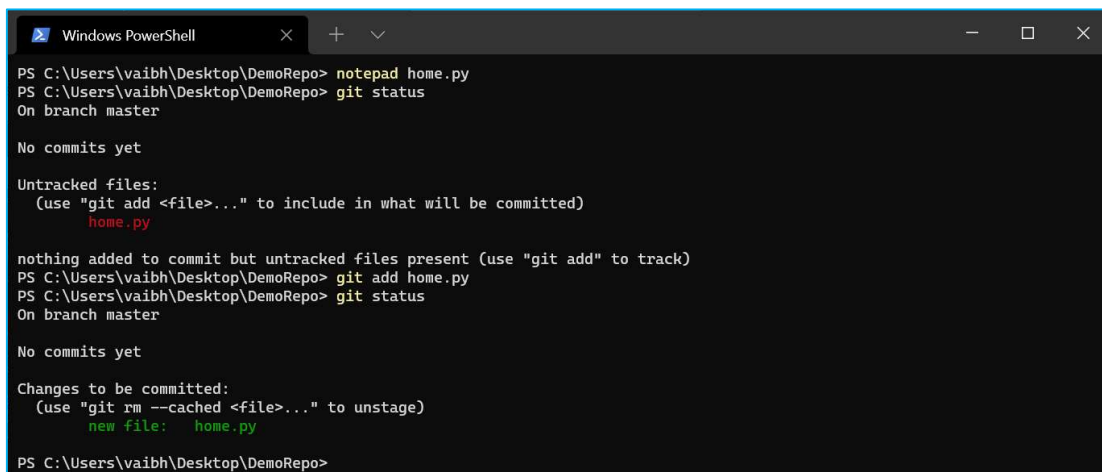
```
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

### ❖ git add [file]

add a file as it looks now to your next commit (stage)



```
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> notepad home.py
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\vaibh\Desktop\DemoRepo> git add home.py
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master

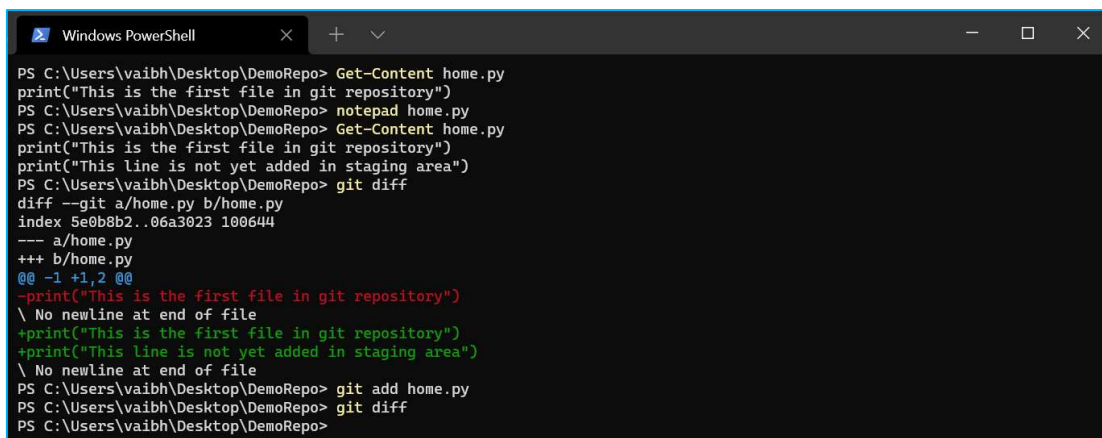
No commits yet

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

PS C:\Users\vaibh\Desktop\DemoRepo>
```

### ❖ git diff

diff of what is changed but not staged



```
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> Get-Content home.py
print("This is the first file in git repository")
PS C:\Users\vaibh\Desktop\DemoRepo> notepad home.py
PS C:\Users\vaibh\Desktop\DemoRepo> Get-Content home.py
print("This is the first file in git repository")
print("This line is not yet added in staging area")
PS C:\Users\vaibh\Desktop\DemoRepo> git diff
diff --git a/home.py b/home.py
index 5e0b8b2..06a3023 100644
--- a/home.py
+++ b/home.py
@@ -1,2 @@
-print("This is the first file in git repository")
\ No newline at end of file
+print("This is the first file in git repository")
+print("This line is not yet added in staging area")
\ No newline at end of file
PS C:\Users\vaibh\Desktop\DemoRepo> git add home.py
PS C:\Users\vaibh\Desktop\DemoRepo> git diff
PS C:\Users\vaibh\Desktop\DemoRepo>
```

❖ **git diff --staged**

diff of what is staged but not yet committed

```

Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master

No commits yet

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

PS C:\Users\vaibh\Desktop\DemoRepo> git diff --staged
diff --git a/home.py b/home.py
new file mode 100644
index 0000000..06a3023
--- /dev/null
+++ b/home.py
@@ -0,0 +1,2 @@
+print("This is the first file in git repository")
+print("This line is not yet added in staging area")
\ No newline at end of file
PS C:\Users\vaibh\Desktop\DemoRepo>

```

❖ **git commit -m "[descriptive message]"**

commit your staged content as a new commit snapshot

```

Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master

No commits yet

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

PS C:\Users\vaibh\Desktop\DemoRepo> git commit -m "My first commit"
[master (root-commit) e2e8d66] My first commit
1 file changed, 2 insertions(+)
 create mode 100644 home.py
PS C:\Users\vaibh\Desktop\DemoRepo> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\vaibh\Desktop\DemoRepo>

```

**INSPECT**❖ **git log**

show the commit history for the currently active branch

```

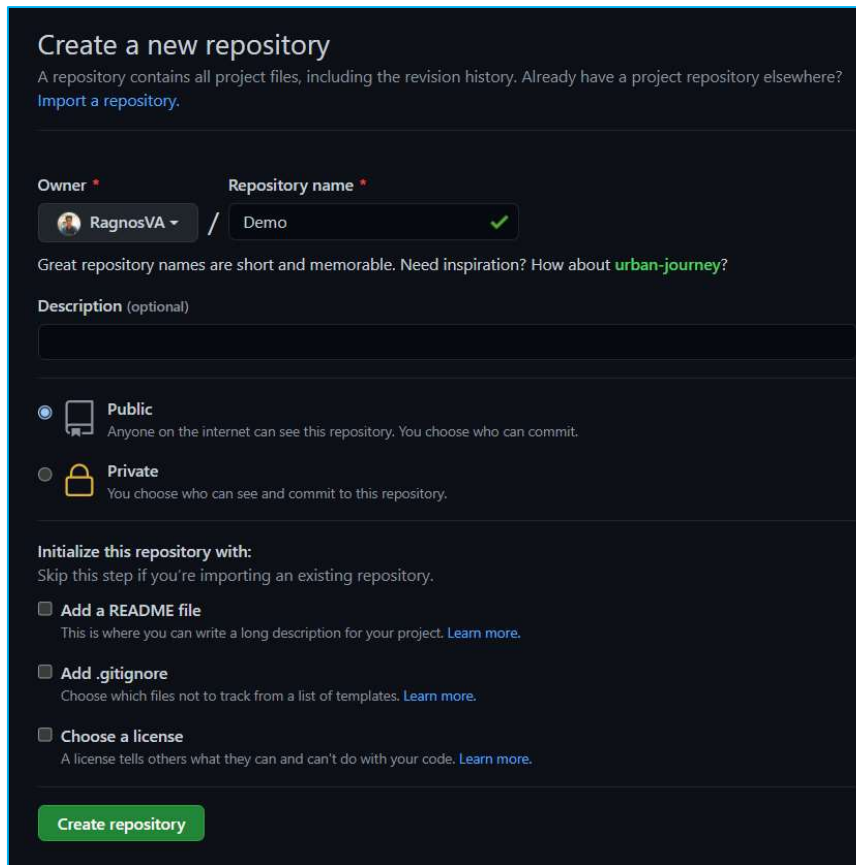
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git log
commit e2e8d667c4250173e042ca787138dfe2f417a022 (HEAD -> master, origin/master)
Author: Vaibhav Ankolekar <76490924+RagnosVA@users.noreply.github.com>
Date:   Tue Aug 10 11:55:47 2021 +0530

    My first commit
PS C:\Users\vaibh\Desktop\DemoRepo>

```

## Adding and Updating Remote repository

First create a remote repository (Github/GitLab/BitBucket etc).



The screenshot shows the 'Create a new repository' page on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, there are fields for 'Owner' (set to 'RagnosVA') and 'Repository name' (set to 'Demo'). A note suggests that repository names should be short and memorable. There is a 'Description (optional)' text area. Under the 'Visibility' section, 'Public' is selected, with a note that anyone on the internet can see the repository. The 'Private' option is also available. Below this, there are three checkboxes for initialization: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each checkbox has a brief description and a link to 'Learn more'. At the bottom, there is a green 'Create repository' button.

❖ `git remote add [alias] [url]`

add a git URL as an alias

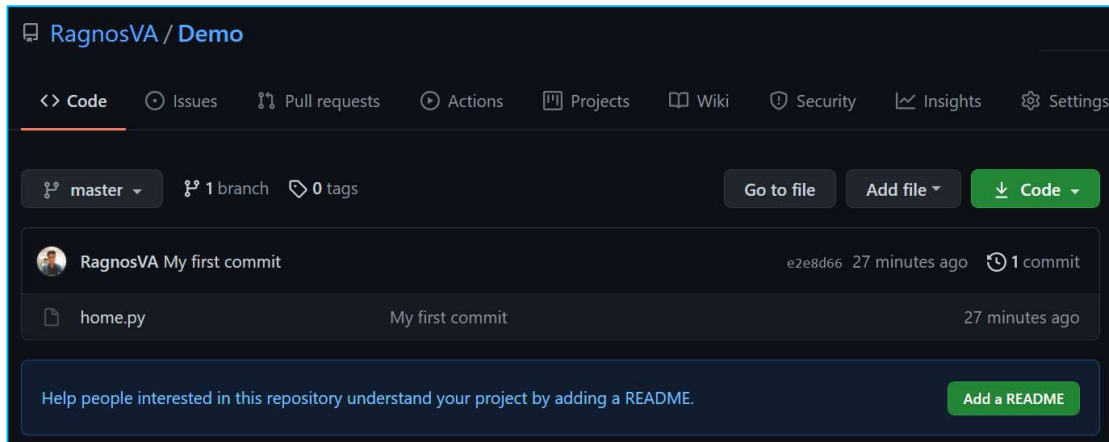
```
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git remote add origin https://github.com/RagnosVA/Demo.git
PS C:\Users\vaibh\Desktop\DemoRepo>
```

❖ `git push [alias] [branch]`

Transmit local branch commits to the remote repository branch

```
Windows PowerShell
PS C:\Users\vaibh\Desktop\DemoRepo> git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 158.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RagnosVA/Demo.git
* [new branch] master -> master
PS C:\Users\vaibh\Desktop\DemoRepo>
```

Refresh the remote repository page to see the changes made in repo.



### Conclusion:

Git provides a way of keeping track of past versions of software and papers, making collaboration between various authors easy, and provides backup for your software. It has proven very useful to the open-source community and in academia as well. Thus, we have successfully performed git operations on local and remote repository.