**1. To create a new local repository:**

Let’s create a directory for repository.

mkdir repo

cd repo

use following command to create repository

git init

It creates a **.git** directory that contains all the Git-related information for your project.

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

2. Create new file **file1.txt** and **file2.txt** in **repo** directory and run following command to check status.

git status

**Status** command displays a list the files you’ve changed and those you still need to add or commit.

A screenshot of a computer

Description automatically generated with medium confidence

**3. Adding files:**

Run following command to add both files:

git add file1.txt file2.txt

Text

Description automatically generated

Application

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Add** command adds one or more files to staging (index).

**4. Commit:**

After staging files, we can commit them into Git. Run following command to commit:

git commit -m "First commit"

A screenshot of a computer

Description automatically generated with medium confidence

you can use -a to commit any files you’ve added with git add, and also commit any files you’ve changed since then.

git commit -a

**Note:** it is dangerous. let’s say you opened a file and changed it by mistake. if you add -a to your commit, all files would be committed and you would fail to notice possible errors.

You can use both -a and -m as well

git commit -am "My commit message"

**5. Further Commits:**

Let’s modify **file1.txt** after first commit. Now to check the changes from the last commit, run following command:

git diff

If you want to have a look at the changes to a particular file, you can run git diff <file>.  
Let’s commit the changes

git commit -am "Second commit"

Graphical user interface, text, application, Word

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**6. To show log:**

To check the history of your project, run the following command:

git log

A screenshot of a computer

Description automatically generated with medium confidence

To view the details of a particular commit:

git show <hash>

Where <hash> is the hex number associated with the commit. you do not need to copy the whole string, and the first 5-6 characters are enough to identify your commit. As in the screenshot, only **2856** is used.

A screenshot of a computer

Description automatically generated with medium confidence

**7. To put code on remote server:**

You could create a project on **GitHub**, **GitLab**, or **BitBucket** and push your existing code to the repository. Conveniently, a remote to which you have write access is called the **origin**.

Run following commands to add a remote origin and then push the code to the origin.

git remote add origin https://github.com/PallaviKatari/.NET.git

git push -u origin master

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Push** command is used to send changes to remote repository.

A screenshot of a computer

Description automatically generated

Git commands related to **Branching** and **Merging**. Branches are used to develop features isolated from each other. It represents an independent line of development. The master branch is the “**default**” branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.

**1. Checkout a repository:**

To create a working copy of a local repository by running the command

git clone /path/to/repository

when using a remote server, your command will be

git clone username@host:/path/to/repository

In our case, as it is public Github repo so simple link is used

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

**2. Branching:**

Let’s create a new branch “**branch1**” and switch on it

git checkout -b branch1

Modify **file2.txt** manually then check **status** and **commit** it.

git status

git commit -am "Branch1 first commit"

To **push** the branch to your remote repository, so others can use it:

git push -u origin branch1

Text

Description automatically generated with low confidence

Graphical user interface, text, application, Word

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

To list all the branches in your repository and to know what branch you’re currently in, run following command:

git branch

To switch from one branch to another:

git checkout <branchname>

Graphical user interface, application

Description automatically generated

**3. Merging:**

Before merging, you can preview the changes using git diff command:

git diff master branch1

If everything looks okay then use following command to to merge **branch1**into your active branch (e.g. **master**)

git merge branch1

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**4. Update your code:**

To update your local repository to the newest commit from the remote repository, run following command:

git pull

**5. Undo Local Changes:**

If you want to sync local code with remote code and drop all your local changes and commits then run following command:

git fetch origin

git reset --hard origin/master

**6. Reset Last commit only:**

If you do wrong commit by mistake and want to remove all changes in last commit, run following command

git reset --hard HEAD~1

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated