# Git terminal commands

Git is called as distributed because , we have both local and remote

Svn doesn’t have local git repo, it will have only remote repo, since git is distributed we have both local and remote

| **Centralized** | **Distributed** |
| --- | --- |
| You can keep changes only in the server | You can keep changes locally (commit) as well |
| Changes can be merged in the server (remote) alone | Changes can be merged locally as well as remotely |

* git clone: Get the complete project from remote to your local machine
* git pull origin <branch\_name>: Get the new changes from remote branch to local branch
* git push origin <branch\_name>: Send your local branch changes to the remote branch
* git remote add <name> <url>: Add a new remote repo link to your local repo
* git remote -v: List all the remote repo URLs linked to your local repo

## set the email after installing git

$ git config --global user.name "First Last"

$ git config --global user.email "myemail@domain.com"

* **git init** adds .git folder and **initializes the current folder to track its changes**
* **git status** displays the current state of the staging area and the working directory, that is, which files are added/removed/modified
* **git diff** **shows the exact changes** with line and column number
* **git add** adds the changes to the staging area. If you have added a new file, this command **starts tracking** the file for modifications.
* **git commit** will **save all the changes** with a unique hash number in the local repository
* **git push** sends the changes to the remote repository (server)
* git log
* git show
* git diff

**HEAD** is a reference variable that always **points to the tip of your current branch, that is, recent commit of your current branch**.

**HEAD** can be used with the following symbols to refer to other commits:

* Tilde symbol (~): Used to point to the **previous commits from base HEAD**
* Caret symbol (^): Used to point to the **immediate parent commit** from the current referenced commit
* git log -2 displays the history of **last two commits**
* git log commit\_id shows the history **starting from commit\_id**
* git log filename displays the list of commits for the file
* git pull is the convenient shortcut key to fetch and merge the content.
  + git pull <remote\_name> <branch\_name>
* git fetch command downloads the remote content to your local repo, **without changing your code changes**.
  + git fetch <remote\_name> <branch\_name> fetches the content from that specific branch in remote to your current working area
* git merge command merges the fetched remote content to the local working tree.
  + git merge <remote\_name>/<branch\_name> merges the content to the specified branch.
* For example: git remote add origin https://github.com/play/repo.git

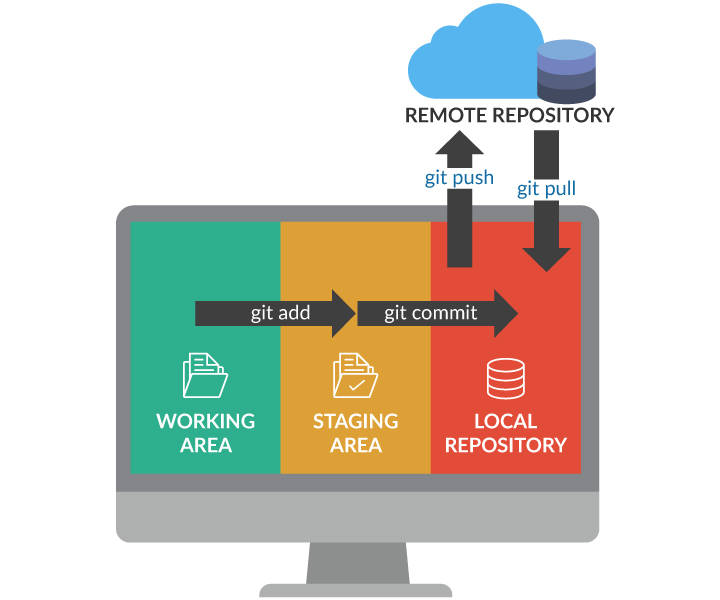
###### Note: Your local repository can be linked to multiple remote repositories as **git remote add**origin1**<url>**, **git remote add**origin2**<url>**

Most used Git commands

**git status**

it will show the files which files are changed on the system

**git add**



All GIT commands

Start .

To initialise a local git repository

git init

To open the folder of current git repo type below command

start .

this start . will even work in windows application also

To pull or check out the code

git clone <https://github.com/manideep-vv/SpringPaintBrushes-1.git>

Git clone <url ends with .git>

This command **downloads the complete project**, all branches, commits and logs from the given remote URL (react repo here) to your local machin

To check for modified files

git status

To see all changes in a tool

Git difftool HEAD

To discard all the changes in the working directory

Git checkout -- <file name>

The above commands will work only for staging area,

I mean it will revert those files present in staging area.

To initialise a local git repo

git init

To add single file to staging area

git add < file name abcd.txt >

To add all files to staging area

git add .

here “.” Means current directory

To remove a file from staging area

git restore --staged "1.unstaged file.txt"

To Commit

git commit -m "first commit"

git commit -m “Initital commit message”

it will commit the files only present in staging area.

After git commit, a unique hash is created and the changes are saved.

Push to upstream

git push

git push origin master

* origin will contain the remote URL
* master is the branch that is pushed (We shall discuss branches later in this course)

To revert all modified files in staging area

git reset –hard

it will revert all the files in staging area only and

It will not impact to any un staged file ,

How to create a git text file using cmd

touch .gitignore.txt

the above will create a text file

to exit from command prompt

:q

To Link local repo to remote repo

* git remote add origin https://github.com/play/repo.git

git remote add origin git@github.com:StephenGrider/docker-react.git git push-u origin master

### Pull fetch

* git pull is the convenient shortcut key to fetch and merge the content.
  + git pull <remote\_name> <branch\_name>

git fetch command downloads the remote content to your local repo, **without changing your code changes**.

*Merge*

* git merge command merges the fetched remote content to the local working tree.
  + git merge <remote\_name>/<branch\_name> merges the content to the specified branch.

Ur friends would have committed to the remote branch, if u want to have all those changes locally ten use git merge

### push to remote branch

echo "# docker-react" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:StephenGrider/docker-react.git

gir remote -v

git push origin master or

git push-u origin master

git remote add origin git@github.com:manideep-vv/ practice-docker-react.git git push-u origin master