.git = All git repo info is stored in this dir
rm -rf ./.git = deletes repo information
.git/config = repo specific config file
~/.gitconfig = global config file
~/.git-credentials = global git user credentials
.gitignore = exclude temporary files or paths
HEAD pointer=most advanced working version
HEAD^ = commit -1 (parent of HEAD)
HEAD^^= commit -2
HEAD~N=commit -N
detached HEAD *= when a specific commit is checked out instead of a branch*

git config = configure git account
local overwrites global which overwrites system
--system = applies to machine and all users
--global = applies to user & global config file
--local = repo specific config (**default**)
--unset variable = remove a variable
-l = list all set git variables
user.name "name" =author of commits
user.email "email" = mail inside commits
git config --global user.name "My Name"
git config --unset user.name

git init *project-name* = create new local repo
git remote add origin url_to_empty_git
git push -u origin master
git clone *url [dir]*=download repo & its history
(***default**: project name= dir)*
*git clone https://**USERNAME**@github.com/rest_of_url*
http[s]://host.xz[:port]/path/to/repo.git/

git status=list new/modified files for commit
--long = status in long format (**default**)
-s = status in short format
-u = untracked files
-uno = tracked files only (***un**tracked **no***)
--ignored = ignored files

git add *pattern*=take file snapshot
-f = force; allow adding ignored files
-A = stage all (modified, new, deleted)
. = stage new & modified, not deleted
-i = activate interactive mode
-u = update; add tracked files, no new files

git commit = record changes to repository
-a=add modified and tracked files automatically
--date "date" = override the commit date
-m "msg" =use "msg" as commit message
--author "name <mail>"= override author info

git diff *options commits/branch*
(default: difference of actual wrt last snapshot)
--cached=last spapshot wrt last local commit
git diff HEAD =diff of actual wrt last local commit
--numstat=added/deleted lines per file
--shortstat=total modified files, added/del lines
--name-only=show names of changed files.
--name-status=names & status of changed files
-b=ignore changes in amount of whitespace
-w=ignore whitespace when comparing lines
--ignore-blank-lines
git diff HEAD^ HEAD → last commit vs previous
git diff topic master → branch topic vs master
git diff topic...master → actual master branch vs topic at the time topic was created

git reset *mode commit*= (**default:** soft ~head)
unstage changes but preserve content
--soft = reset head to commit; preserve local changes; leave files as git status would put it
--hard = Resets the index and working tree to *commit*; Any changes to tracked are discarded.

git push *[remote] [branch]*=Update remote repo with committed snapshots (**def** origin current**)**
-all=push all branches
-f= force; can cause remote to lose commits
-u= add upstream (tracking) reference, used by argument-less git-pull and other commands
--delete *[remote] [branch]*= *[remote] :[branch]*
***git push origin master**=push master to update master in origin repo. (if necessary create new branch)*

git pull=download & merge changes (***fetch+merge)***
--ff= fast forward, only update branch pointer, without creating a merge commit. (**default**)
--no-ff= create a merge commit
--ff-only= refuse to merge if ff is not possible
-r, --rebase[=false|true|preserve]
--all= Fetch all remotes.
-f = override check when updating the branch
***git pull origin next** = merge into current branch the remote branch next*

git fetch= Download history from the repo. Keep local branch up to date wrt remote. It never changes local branches and is safe to do without changing your working copy.

git merge *[bookmark]/[branch]*= Combines bookmark's branch into current local branch

git-rebase *[branch]*=Fw local commits from branch

git branch = list all local branches
[name] = Create new branch (no swithching)
-f *[name]*= force the creation of new branch
-a = list remote-tracking and local branches
-r = list remote-tracking branches
--merged=list branches fully contained by HEAD
-d *[name]*= del a fully merged branch
-D *[name]*= force de no warning necessary
--edit-description= edit branch description
-m *[name]*= rename current branch to name
-M *[name]*= force rename (overwrite existing)
-u = set-upstream so that pull merges
--unset-upstream = remove upstream config
-u = set-upstream so that the pull merges
-v = verbose; show more info for each head
***git branch --va** = list branches with more info*
***git branch -u upstream/foo** = set upstream while on local foo branch*
***git branch -u upstream/foo foo**= set upstream while not on* foo local branch:

git checkout *[branch/commit]*= determines which revision of your project you want to work on. → Switches to branch/commit and updates the working directory
***git checkout** 56a4e5c08 File.name*
-b = create new branch and switch
-B = create or reset branch if exists
=git branch -f + git checkout
-f = force switch/ignore unmerged entries
--detach =detach the HEAD
git checkout 56a4e5c08 → edit old commit
git checkout -b test-branch → save work
git checkout -b test-branch 56a4e5c08 → correct
git checkout master; git checkout -d test-branch

git remote = manage set of tracked repos
add= add a new remote
prune= delete all stale tracking branches
rename=rename & update its tracking branches
rm=remove a remote and its tracking branches
set-url=change URL for a remote
show= show information (**default**)
update=fetch updates for a set of remotes
-v=show remote url after name

git stash=temporary store modified tracked files
git stash list=lists all stashed changesets
git stash pop=restores most recent stashed files
git stash drop=discard most recent stashed set

git rm *[file]*=delete & stage file for deletion
rm file + git add file
git rm --cached *[file]*=remove from tracking but preserves file locally
git mv *[name] [new]*=mv & prepare to commit

git log=version history of current branch
--abbrev-commit = show short commit hash
--shortstat=generate short summary
--summary= extended condensed summary
--oneline= pretty=oneline --abbrev-commit
--name-only= show names of changed files
--name-status= changed file name status A/D/M
--numstat=show num of add/del lines
--format: https://git-scm.com/docs/pretty-formats
--reverse=display commits in reverse order
--all= show all commits from all branches
--diff-filter=*A/D/M* =select certain kinds of files
--follow *[file]*= file history with renames
--no-follow *[file]*=do not follow renames
--author *[name_pattern]*
--grep "log *text pattern*"
--children= display children of commit
--parents= display parents of commit
--graph=graphical representation
--merge=after failed merge, show conflict files
--merges=display only merge commits
--no-renames=turn off rename detection
git log --diff-filter=D --summary | grep del → list del files

git show *[commit]* = Outputs metadata and content changes of the specified commit

git ls-files *--other --ignored --exclude-standard* = *Lists all ignored files in this project*

git show-ref --head --abbrev= all refs with head

Cancel uncommitted snapshots: git reset file
Reset file to last commit: *git checkout HEAD file*
Reset to remote master & update files: git reset --hard origin/master
Download latest from remote: git fetch –all remote
Change Repo Name:
1) Change the repo name at webpage
2a) git remote set-url origin
2b) git remote rm origin
git remote add origin new_url

gitblame
gitk

https://github.com/git-tips/tips