|  |  |  |
| --- | --- | --- |
| **Setup** | | |
| 1 | Command to check config options | git config --list git config --list --show-origin git config user.name |
| 2 | Command to check config options + file origin | git config –show-origin user.email |
| 3 | Command to configure config options  name and email | git config --global user.name "Akintunde Adegbayo" git config --global user.email akt231@yahoo.com |
| 4 | Command to configure config options  editor | git config --global core.editor emacs |
| 5 | Command to prepare a directory for git | #make sure you are in current directory |
| 6 | Command to open git config file in editor of choice | git config --global --edit |
| **Queries** | | |
| 1 | Command to Check Git Status;  ***Checks what branch you are on***  ***compares branch you are on to remote branch***  ***lists untracked files***  ***lists modified tracked files*** | git status  git status -s ***#concise version*** |
| 2 | the difference between your working copy and the index | git diff |
| 3 | what have you staged that you are about to commit? | git diff --staged |
| 4 | see difference between the state of files in the index/stage-area and HEAD, which is typically the last commit on the branch you're working on | git diff --cached |
| 5 | Command for git help | git <verb> -h ***#concise version*** git <verb> help ***#verbose version*** git help <verb> ***#verbose version*** |
| 6 | Command to check content of a directory | ls -la ls |
| 7 | Command to check current file path | pwd |
| 8 | To view Commit History | git log |
| 9 | To view Commit history and show difference (patch output) | git log -p  git log --patch |
| 10 | To view Commit history and show difference (patch output) and show only last 2 entries | git log –patch -2 |
| 11 | To view commit history in one line | git log –oneline  git log --pretty=oneline -5 |
| 12 | create a new file in working directory not yet tracked | touch file.txt |
| **staging operations** | | |
| 1 | add new and changed items to staging area | git add <file\_name>  git stage <file\_name> |
| 2 | add all new and changed items to staging area | git add \*  git stage \* |
| 3 | undo a addition of new and changed items to  staging area | git reset HEAD <file\_name> |
| 4 | to discard changes to a file | git checkout -- <file> |
| 5 | to unstage a staged file | git restore –staged <file> |
| 6 | to restore file to last commit | git restore <file> |
| 7 | create a new file in working directory not yet tracked | touch <newfile> |
| **commit operations** | | |
| 1 | commit changes to files added before | git commit -m "comment\_descr." |
| 2 | commit changes to files and automatically add them as well | git commit -am "comment\_descr." |
| 3 | stages files automatically before a commit can be used instead of git add. Adds git add on any file that has been tracked-in last commit and modified, only works on tracked files | git commit -a m "comment\_descr." |
| 4 | stages files automatically before a commit can be used instead of git add. Adds git add on any file that has been tracked-in last commit and modified, only works on tracked files | git add .  git commit -m “Commit Message” |
| 5 | stages files automatically before a commit can be used instead of git add. Adds git add on any file that has been tracked-in last commit and modified, only works on tracked files | git commit -am “Commit Message” |
| 6 | stages files automatically before a commit can be used instead of git add. Adds git add on any file that has been tracked-in last commit and modified, only works on tracked files | git commit -a -m “Commit Message” |
| 7 | undo a commit | git commit --amend |
| **remote operations** | | |
| 1 | to see your remote repos | git remote |
| 2 | to see your remote repos url and list all remotes | git remote -v |
| 3 | Inspecting a Remote Repo | git remote show <remote>  git remote show origin |
| 4 | change a remote’s shortname | git remote rename <oldname> <newname>  git remote rename pb paul |
| 5 | reove the remote repo of a local repo | git remote remove <shortname>  git remote remove paul |
| 6 | to add a new remote repo | git remote add <shortname> <url>  git remote add pb https://github.com/plbne/ticgit |
| 7 | update your local copy of the repository but *dont modify any files in the working directory* | git fetch |
| 8 | updates your local copy of the repository and then merge the changes from the remote repository into your current branch and working directory. | git fetch  git merge FETCH\_HEAD  or  git pull |
| 9 | push to github | git push |
| 10 | push to remote repo , creates a **local branch** in the remote repository, including all specified commits and objects | git push <name of remote server> <branch name>  $ git push origin master |
| 11 | adds all local branches to the indicated remote repository | git push --all |
| 12 | push to github | git push |
| **Remove a file from git** | | |
| 1 | Removed file from git  you have to remove it from your tracked files (more accurately, remove it from your staging area) and then commit. Deletes file physically too | git rm -f <file name> |
| 2 | Removed file from git staged area but lists file as to be tracked | git rm –cached <file name> |
|  | As an example, if you commit and then realize you forgot to stage the changes in a file you wanted to add to this commit, you can do something like this:  You end up with a single commit — the second commit replaces the results of the first. | $ git commit -m 'Initial commit'  $ git add forgotten\_file  $ git commit --amend |
|  |  |  |
| **TAGGING** | | |
|  | List Tags in repo | git tag |
|  | Create a tag on a commit | git tag -a <tag> -m "tag messge"  git tag -a v1.4 -m "my version 1.4" |
|  | To check specific tag data | git show <tag>  git show v1.4 |
|  | to add a tag at a specific commit | git tag -a <tag> <commit hash>  ie  git log --pretty=oneline -5  git tag -a v1.2 9fceb02 |
|  | push tag to remote repo | git push origin <tagname>  git push origin v1.5 |
|  | push all tags to remote repo | git push origin --tags |
|  | delete specific tag | git tag -d <tagname>  git tag -d v1.4-lw |
|  | delete a remote tag | git push origin --delete <tagname> |
|  |  |  |
|  |  |  |
| **Branches** | | |
| 1 | create a new branch called testing | git branch <branchname>  git branch testing |
| 2 | switch to an existing branch called testing | git checkout <branchname>  git checkout testing  git switch <*branchname>*  git switch testing |
|  | create a new branch and switch to it | git switch -c testing  or  git switch –create testing |
|  |  | git switch *branch* |
|  |  |  |

Stopped @

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging