|  |  |  |
| --- | --- | --- |
| **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 |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
|  |  |  |
| **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 |
|  | what have you staged that you are about to commit? | git diff --staged |
|  | 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 |
| 3 | Command for git help | git <verb> -h ***#concise version*** git <verb> help ***#verbose version*** git help <verb> ***#verbose version*** |
| 4 | Command to check content of a directory | ls -la ls |
| 5 | Command to check current file path | pwd |
| 6 | To view Commit History | git log |
| 7 | To view Commit history and show difference (patch output) | git log -p  git log --patch |
| 8 | To view Commit history and show difference (patch output) and show only last 2 entries | git log –patch -2 |
| 9 | To view commit history in one line | git log --oneline |
| 10 | create a new file in working directory not yet tracked | touch file.txt |
| **add new and changed items to staging area for tracking** | | |
| 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 |  |  |
| 5 |  |  |
| 6 | create a new file in working directory not yet tracked |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| **commit item** | | |
| 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 |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
|  |  |  |
| **push to github** | | |
| 1 | push to github | git push |
| 2 | creates a **local branch** in the remote repository, including all specified commits and objects | git push <name of remote server> <branch name> |
| 3 | adds all local branches to the indicated remote repository | git push --all |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| **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> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Branches** | | |
| 1 |  | git checkout branch |
| 2 |  | git switch *branch* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Stopped @

https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History