**GIT CHEATSHEET – Quick Guide for Git Commands**

|  |  |
| --- | --- |
| **Git Commands** | **Description** |
| Git branch “branch\_name” | Creates a Branch |
| Git branch –D “branch\_name” | Deletes a branch |
| Git checkout “branch\_name” | Switch to a branch |
| Git branch --all | Show all branches |
| git checkout - | Switch to previous branch |
|  |  |
| Git add . | Add files to the local index(staging area) |
| Git commit --m “Commit message” | Commit message and files |
| git reset --hard master  git reset --hard head^  git reset --hard head~25 | Moves a branch from current position to master  ^ caret basically means a back step  ~25 tilde means 25 steps back |
| Git restore --staged “file\_name”  Git restore “file\_name” | to unstage files from staged area  Discard changes in working directory |
| Git checkout -- “file\_name” | Completely threw away any modifications |
| Git stash | record the current state of the working dir & want to switch to clean dir |
|  |  |
| Git commit --am “add and commit message” | Add to stage & commit message together |
| Git checkout “target\_branch”  Git merge “source\_branch” | Merge source branch to target branch |
|  |  |
| git checkout HEAD^ | 'detached HEAD' state |
| git branch “branch\_name” “commit\_number” | Create branch with detached state & new changes |
| git log --oneline --graph --decorate --all | Display history of all branches |
|  |  |
| git cherry-pick edf6319 | fixed a bug in a branch, and then apply the same fix on top of another branch |
| git pull origin master =  [ git fetch  git status  git merge origin master ] | Git pull = git fetch + git merge |
| git push -u origin “branch\_name” | Pushing a new branch to the remote |
| git config --global alias.unstage 'reset HEAD --'  git unstage “File\_Name” | Aliasing |