

<<<1. Setup and Init>>>

git init: initialize & set up the tools necessary to start

git clone [url]: clone an entire repository

git config ... : configure user info GOOGLE IT

<<<2. STATUS CHECK, Staging, and Committing>>>

git status: check the status of changes being made

git diff: check what is changed but not staged

git diff --staged : diff of staged files

git log: check the commit history

git add [file]: add file to the stage

git reset [file]: un-stage file, but keep the changes

git commit -m "message": Commit the changes

<<<3. Branching and Merging>>>

git branch: list out all the branches. The branch that user is on is indicated (*)

git branch [branch_name]: create a new branch with the name

git branch -d [branch_name]: delete the branch

git checkout [branch_name]: switch to the other branch. all changes must be dealt with first

git merge [branch_name]: will merge the specified branch to the current branch

<<<4. Inspection>>>

git log: shows the commit history to the branch

git log --follow [filename]: show commits that changed the file

git reflog: show the changes being made to HEAD. aka entire history of what is going on

git show [SHA]: show anything in Git in human-readable format (SHA could be HEAD)

<<<5. Sharing & Update>>>

git remote ...: makes remotes for connecting with others. GOOGLE IT

git fetch [alias]: brings down all the branches (most likely to origin)

git merge [alias]/[branch]: merge remote branch

git push: move all commits to the remote repo branch

git pull: fetch and merge from the remote to the local branch

<<<6. Undo Changes & Rewrite History>>>

git rebase [branch]:

git reset [commit]: Content stays on the disc, but commit history is lost! :((safer)

git reset --hard [commit]: Clear staging area and rewrite working tree (CAUTION)

git revert [commit]: Revert one specific commit without erasing all past commits subsequent

<<<7. Stashing (Advanced)>>>

git stash: temporarily store the working directory/changes on a stack

git stash list: list out all the stashed files in stack-order

git stash apply: Default take from the top, but you can specify 'stash@{#}' for specific stash

git stash pop: Apply the top and also drop it

git stash drop: Discard the top of the stash

