Git Commands:

- git clone *gitRepo* # Clone gitRepo to local machine.
- git status
- git add * : Stage unstaged files
- git log
- git log -*n* : log most recent n items (commits)
- git commit -m "*Commit comment*" : Commit staged files with a commit comment
- git cherry -v : View commit that aren't pushed yet
- git difftool *CommitSHA1*^! : Use gui difftool to view diff between commit and parent
- git difftool *CommitSHA1~ CommitSHA1* : View diff made by commit
- git cat-file -p *SHA1*: views content of SHA1
- git branch
- git branch -r : list remote branches
- git branch -a : list local and remote branches
- git checkout -t <name of remote (origin)>/*branchName* : Checkout remote branch
- git branch *branchName* :
- git branch --all : Show all branches including branches on *remotes/origin*
- cat .git/HEAD: Show object that Head is currently pointing.
- git checkout *branchName* : Move Head and update working area
- git checkout -b *newBranchName* : Create & checkout new branch from current branch
- git push -u origin newBranchName : Push new branch to origin
- git branch -d *branchName* : Delete local branch if clean
- git push origin --delete *branchName* : Delete remote branch
- git remote prune origin : Cleanup deleted remote branches
- git merge *sourceBranchName* : Merge files from *sourceBranchName* to current branch
- git mergetool: Lanches GUI merge tool (eg Araxis) to resolve conflicts after merge
- git rebase master: Copy all commits of current branch since last shared commit and add them on top of the master commit history. Moves current branch head to copied commits
- git push : Push local commits to remote
- git pull : Perform git fetch / git merge
- git diff : Compare Working Area to Index (Staging Area)
- git diff --cached : Compare Index to local Repository
- git stash list: Lists data in Stash
- git stash clear : Clear data in Stash
- git rm fileName : Delete unstaged file from the local area
- git clean -n : List all files/directories that will be cleaned
- git clean -fd : Delete untracked files from working area (be careful!)
- git fetch --tags origin
- git merge *tagName* : Merge branch up to tagName
- git checkout tags/<tag_name> -b <branch_name>
- git cherry-pick -x <commit-hash> : Cherry-pick commit and keep original work items, etc.
- git checkout source_branch <paths>... : Copy files from source_branch to current branch
- git log --left-right --graph --cherry-pick --oneline <branch1>...<branch2> : List commit diff
- git branch -r --contains <commit> : List branches that contain commit

- %history : List previously executed commands
- %!# : Run comand # (example %!86)
- git checkout <source_branch> <path/to/new_file> # copies new_file from source_branch
- git gitCommand &> gitCommandOutput.txt # redirect output of git command to file

Four Areas: (Stash:Working Area:Index(Staging Area):Repository):

Moving Data to the Right

- git add *fileName* : Move update from Working Area to Index
- git commit -m "*Commit comment*" : Move update from Index to Repository with comment
- git mv fileNameOld fileNameNew : Rename and stage file
- git stash apply : Moves stash@{0} state to Working Area and Index (does not delete)
- git stash pop : Moves stash@{0} state to Working Area and Index and delete stash@{0}
- git stash pop *stashID* : Moves stashID's state to Working Area and Index and deletes it

Move Data to the Left

- git checkout *branchName* : Move commit of branch to Index and Working Area
- git rm --cached *fileName* : Move file from Index to Working Area (unstage file)
- git reset --hard *SHAValue* : Moves branch Head to SHAValue's commit in Repository, reverts Index and Woking Area to status of SHAValue's commit (child commits are lost)
- git reset (--mixed) *SHAValue* : Moves branch Head to SHAValue's commit in Repository, reverts Index to status of SHAValue's commit (child commits are lost)
- git reset --soft *SHAValue* : Moves branch Head to SHAValue's commit in Repository, does not change updates in Working Area or Index (child commits are lost)
- git reset (--mixed) HEAD : Unstages updates in Index
- git reset --hard HEAD : Unstages updates in Index and deletes them from Working Area
- git reset (--mixed) HEAD *fileName* : Unstages fileName
- git checkout HEAD *fileName* : Undo changes in fileName in Working Area and Index
- git stash --include-untracked : Moves all updates in Working Area to Stash and checkout current branch (Moves branch state to Working Area and Index)
- git stash save *myMessage* : Moves tracked updates in Working Area to Stash and checkout current branch with "*myMessage*" (Moves branch state to Working Area and Index)

Solving Conflicts

- git merge *branchName* : Merge branchName files to current branch -> "merge failed"
- git status : Display unmerged paths -> Need to edit files to resolve conflict & add to Index
- git mergetool: Lanches GUI merge tool (eg Araxis) to resolve conflicts after merge
- git add *fileNameWithConflict* : Tells git that conflict has been resolved & ready to commit
- git commit : Complete merge with commit

History

- git log --graph --decorate --oneline : Display readable commit history of current branch
- git log --patch : Show details of each commit
- git log --grep *text* --oneline : Show all commit with text in comment message

- git log -Gtext --patch : Show everywhere that text is in commit
- git log *branch0..branch1* --online : List commits in branch1 that aren't in branch0
- git show HEAD : Show information about commit of Head
- git show HEAD^ : Show information about parent commit of Head
- git show HEAD^^ : Show information about parent of parent commit of Head
- git show HEAD~2 : Show information about parent of parent commit of Head
- git blame *fileName* : Show file change history by author
- git diff HEAD HEAD~2 : Show diff between current commit and two parents ago
- git diff *branch0 branch1* : Compare two branches
- git difftool *branch0 branch1* : Launch GUI to compare two branches
- git difftool *branch0 branch1* -- *path/subpath* : Compare path of two branches
- git commit --amend : Copy previous commit add create updates. Old commit is lost
- git rebase -i commitRef : Interactive edit history from commitRef's commit forward
- git revert *commitSHA* : Create commit opposite changes from commitSHA's commit
- git reflog : Show reflog