git --version

git --help

git config --list

git config --global user.name "okalu"

git config --global user.email "okalu@miu.edu"

git init

git status

git add hello.txt (or git add .)

git commit -m "First cut"

git log

git remote add origin https://github.com/okalu/git-repo1.git

git remote (or git remote -v)

git branch

git push -u origin main

git diff hello.txt (see diff btw changes in working area and committed changes)

git config --global alias.st status (creates an alias)

git checkout <branch-name> or <commit-hash> (switch branches or restore working tree files)

git switch <branch> switch to a specified branch

git tag -a <tagname> -m <descrip-msg> (add or list or delete or verify a tag - marker on a specific commit)

git restore hello.txt (revert changes in the working directory - changes made without exec of git add)

git restore --staged hello.txt (undo changes made after exec of git add)

git revert HEAD (undo the latest committed changes)

git reset <commit-hash> (set the current branch HEAD to <commit>i.e. removing commits from the branch)

git commit --amend (update the latest commit without creating a new one - deletes existing commit and replaces it with the new/modified one)

git branch <new-branch-name> (creates a new branch)

git branch -d <branch-name> (deletes an existing branch)

git merge <branch-name> (Merges changes from <branch-name> into current active branch

What is a merge conflict? - (e.g. A merge conflict will happen when the exact same line is changed in separate branches)

Create a merge conflict and take a screenshot of it

Resolve the merge conflict - (1. Open and edit the conflicted file, removing the conflict indicator characters; save the changes; do git add <filename>; do git commit -m <msg>