Git commit,

Git Branch Name: Include the current branch and all parent commit in a new branch

Git checkout Name: Move to the branch “name”.

#Git checkout -b[name]: create and check out at same time.

Git merge: a commit with two parents. (git checkout master; git merge bugFix)

Git branch -d name: delete branch

Git rebase master: create a copy into the master in sequence.

Detach HEAD from branch and attach it to the commit.

Git log to see commit hash

Git checkout HEAD^ : move head to parent

Git branch -f master HEAD~3 : move the master branch to three parents behind HEAD.

Git cherry-pick: select commit you want to add to current branch.

Git rebase -i HEAD~4: interactive rebase dialoag opens

Git rebase -i: re-order the commits

Git Commit --amend: make modification to a commit

Git tag v1 c1: add a permeant tag to c1

Git describe: path to last tag

Shift + : and WQ : to add massage when commit.

Git add . :add current folder

Git commit -a: commit the change file.

Git –help: help document.

### …or create a new repository on the command line

echo "# Reference" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/llone0031/Reference.git

git push -u origin master

**Example: Contribute to an existing repository**

# download a repository on GitHub.com to our machine

git clone https://github.com/me/repo.git

# change into the `repo` directory

cd repo

# create a new branch to store any new changes

git branch my-branch

# switch to that branch (line of development)

git checkout my-branch

# make changes, for example, edit `file1.md` and `file2.md` using the text editor

# stage the changed files

git add file1.md file2.md

# take a snapshot of the staging area (anything that's been added)

git commit -m "my snapshot"

# push changes to github

git push --set-upstream origin my-branch

**Example: Start a new repository and publish it to GitHub**

First, you will need to create a new repository on GitHub. You can learn how to create a new repository in our [Hello World guide](https://guides.github.com/activities/hello-world/#repository). **Do not** initialize the repository with a README, .gitignore or License. This empty repository will await your code.

# create a new directory, and initialize it with git-specific functions

git init my-repo

# change into the `my-repo` directory

cd my-repo

# create the first file in the project

touch README.md

# git isn't aware of the file, stage it

git add README.md

# take a snapshot of the staging area

git commit -m "add README to initial commit"

# provide the path for the repository you created on github

git remote add origin https://github.com/YOUR-USERNAME/YOUR-REPOSITORY.git

# push changes to github

git push --set-upstream origin master

**Example: contribute to an existing branch on GitHub**

# assumption: a project called `repo` already exists on the machine, and a new branch has been pushed to GitHub.com since the last time changes were made locally

# change into the `repo` directory

cd repo

# update all remote tracking branches, and the currently checked out branch

git pull

# change into the existing branch called `feature-a`

git checkout feature-a

# make changes, for example, edit `file1.md` using the text editor

# stage the changed file

git add file1.md

# take a snapshot of the staging area

git commit -m "edit file1"

# push changes to github

git push