- Create branches

- Switch between branches

**- Commit to branches**

- Revert commits

- Delete branches

- Merge branches

- Revert merges

- Clone branches

- Push branches

- Pull branches

- Stash branches

**Create a branch without switching:**

Git branch <branch\_name>

**Create a branch and switch to it:**

Git checkout -b <branch-name>

**Switching to a branch:**

Git checkout <branch-name>

**Commit to branches:**

First checkout your new branch, or existing one. Then add the file and lastly add your commit message.

Git checkout <branch-name>

Git add <files>

Git commit -m <message>

**Revert commits:**

Git status (make sure it’s clean)

Each commit has unique hash (2f5451f f.e.x)

Git log (or) git log –oneline

Git revert 2f5451f (Find the hash you want to revert)

To prevent git from asking for a commit message, git revert 2f5451f –-no-edit

**Clone Branches:**

Git clone -b <branch-name> <remote\_repo>(github website)

To make sure you’ve cloned correctly:

Git branch

**Push Branches:**

Git push <remote-repo> <branch-name>

This is mostly used to publish an upload local changes to a central repository.

Also used to share modifications with remote team members.

**Pull Branches:**

Git pull <remote-repo>

Git merge origin/<current-branch>

Used to download content from the specified remote repository.

Then you merge the remote content refs and heads into a new local merge commit.

**Stash Branches:**

Save changes to branch A.

Git stash

Check out branch B

Do whatever you need to do in branch B

Commit and push to remote

Check out branch A

Git stash pop (to get your stashed changes back).