Github Summary

1. git init -b main — creates a ew repo and names the def branch main
2. Git status
3. Git add . Git status
4. Git commit -m “this is how your first commit will be”
5. Git log. status (commit without staging git commit -am””)
6. Git branch <branchName> to create a new branch
7. Git branch — to confirm the branches we have
8. Git checkout <branch-name> to switch to another branch
9. Git status. git add .

// Changing master to main

1. Git checkout master then git merge emergency-branch
2. Git branch -d emergency-branch

Github

- Git remote set-url origin https://<githubtoken>@github.com/<username>/<repo>.git
- Git remote -v verifies the new remote url
- Git push -u origin main
- Git pull origin. — pulls changes from master branch and merges into the local repo
- Git push origin
- Git pull — pulls changes from all branches of the remote repo
- Git branch -a to see all local and remote branches
- Git checkout -b new-branch — to create a new branch in the local repo
- git push origin update-readme —pushing this branch to remote repo
- Git clone https://<username>:<githubToken>@github.com/username/repo.git
- Git log —online

Reverting back

- Git revert HEAD — to revert to the latest commit
- Git revert HEAD-x (1 go back 1 more, 2 go back 2 more) to revert to earlier commits

- git revert requires the id of the commit **you want to remove** keeping it into your history
- git reset requires the commit **you want to keep**, and will consequentially remove anything after that from history.

- git revert <insert bad commit hash here>. — then reverts this commit and goes the commit just before this
- https://stackoverflow.com/questions/19032296/how-to-use-git-revert

- How to use Git Revert - Stack Overflow
- git reset 01b56c6. — brings back files to the specified commit but deletes all commit after that
- Git revert hash -m “revert with custom message”
- Git restore —staged to unstage changes that have been added to the staging area