Git Basics

Antwane Mason
Yuriy Stejko

Creating Repo

- Local creation
  - **git init**
    - going to make a .git subdirectory
    - call this command in an empty folder that will house your repo
- Register repo info
  - **git config [--global] <option> “<value>”**
    - global uses <option> info across all local repos
    - options include following: user.name, user.email, core.editor

Creating Repo (continued)

- Remote repo creation
  - Use github link to request Free Student Pack: https://education.github.com/discount_requests/new
  - After creating github account, use “create repository” link to create remote repository
  - Click “clone or download” button on repository page to get repo link
  - perform following command in local directory
    - **git clone <link>**
- Important Notes
  - **NEVER** post homework or project solutions to public repository unless given permission by instructor
  - Use private repositories to work on homework and projects when you cannot publish publically
  - Can use cloud storage (e.g. Dropbox, OneDrive, Google Drive) and create local repo in synced folder on PC
  - **To get powerpoint presentation: git clone https://github.com/Admasnd/git-workshop.git**

Committing Changes

- Three areas: working directory, staging area, repo
- Adding files to staging area
  - **git add <file>**
- Commit
  - **git commit [- m “<message>”]**
    - -m allows you to type message inline
    - adds snapshot of files in staging area to repo

Syncing with Remote Repo

- Pulling changes from repo
  - **git pull <remote>**
    - Ex. git pull origin
- Pushing changes to repo
  - **git push <remote>**
    - Ex. git push origin

Revert Changes

- Changes in working directory
  - **git reset --hard**
    - removes uncommitted changes since last commit
    - impacts working dir and staging area
  - Good ol’ fashion delete from OS
- Changes in staging area
  - **git reset <file>**
    - remove file from staging area only (not working dir)
  - **git reset**
    - reset staging area to reflect last commit
    - does not change working directory

Revert Changes (continued)

- Changes in repository
  - **git revert <commit>**
    - new commit created reverting changes introduced by <commit>
  - **git checkout <commit> <file>**
    - replace <file> in working directory with version of <file> in <commit>
- Viewing Change Related Info
  - **git log --oneline**
    - show commit ids and titles for past commits
  - **git checkout <commit>**
    - state of repo changed to <commit> except local modifications (staged & unstaged)
    - used to view state of files at a particular commit

Making Branches

- Applications
  - try out experimental changes
  - focus a series of commits on a feature
- List branches
  - **git branch**
- Create a branch
  - **git branch <branch>**
- Delete a branch
  - **git branch -d <branch>**
- Go to a branch
  - **git checkout <branch>**

Further Reading

- Stashing local changes: https://www.atlassian.com/git/tutorials/git-stash/

Bibliography

1. https://git-scm.com/book/en/v2/Getting-Started-Git-Basics
2. https://www.atlassian.com/git/tutorials/