# CICS 191 S19

## Using Git

# Today's class

- Download and install git
- Create git account
- Create a local git repo
- Create a file and add it to the local git repo
- Edit file locally and update the file on git
- Learn about branches **we won't create a branch today
- Clone someone's repo
- Assignment: add a readme file to your repo, clone a repo, add a file and push to git

# Download and install git

- Go to https://git-scm.com/downloads and download git for your OS
- Run the downloaded installer to install git
  - Check installation using this command: git --version
- Create a github account here https://github.com/join
- Configure git:
  - Configure username: git config --global user.name "your_username"
  - Configure email: git config --global user.email <your_email_address@example.com>
  - Check configuration using: git config --global --list

# Create a local git repo

- Create a local git repo
  - Navigate to your preferred location (e.g. Desktop, Documents) and create a folder to hold your project
  - Create a folder named “myproject”
    - You can also create on it terminal using:
      - cd ~/Desktop/
      - mkdir myproject
  - Let’s access the project folder from terminal:
    - cd ~/Desktop/myproject
      - You can check the folder path by using: pwd
  - Let’s initiate a git repo
    - Type on the terminal: git init
      - If successful you should see this response: Initialized empty Git repository in /Users/<your_username>/Desktop/myproject/.git/

# Create a file and add it to the local git folder

- Create a file
  - Open your text editor (e.g. MS Word, textEdit) and create a new blank document
  - Type in your document a description of your project. e.g. “my project will collect and analyze picture of dogs and cats”. Feel free to style it as you please.
  - Save the file as “about_my_project.txt” inside the “myproject” folder i.e. ~/Desktop/myproject
  - Check that your file is in the myproject folder:
    - On your terminal type:
      - cd ~/Desktop/myproject
      - Ls
    - If the file was saved in folder you should see it listed.

# Add file to local git repository

- We need to make sure that git recognizes the changes (i.e file) we've made to the repo.
- We check that using this command:
  - git status
  - We should see a response such as:
    - On branch master. Initial commit. Untracked files:(use "git add <file>..." to include in what will be committed). About_myproject.txt. nothing added to commit but untracked files present (use "git add" to track)
- We add the file to git using this command: git add about_myproject.txt
  - Generally, to add a file to the repo we use the command: git add <filename>
- We can check whether the file was added by checking the repo status as before
  - On your terminal type: git status
  - This time your response should say that some changes need to be committed e.g. Initial commit. Changes to be committed:.  (use "git rm --cached <file>..." to unstage)

# Commit changes to local repo

- Commit changes to the local repo
  - We use this command to commit the changes to out local repo:
    - git commit --m “initial commit. Added the about_project file”
      - Generally, when commiting changes you use the command: git commit “Your commit message”
      - Your message should be help you know what you did in that commit, i.e. what changes youy made
    - If commit successful, you should see a message such as:
      - [master (root-commit) b345d9a] “Initial commit. Added the about_project file”
      - 1 file changed, 1 insertion(+)
      - create mode 100644 about_myproject.txt
    - We can check the status again if we want using:  git status

# Create a remote repo

- Everything we have done so far is on our machine. We want our code to be available outside our machine, on the remote server hosted by Github.
- So we have to create a remote repository. To do that:
  - We got to Github at github.com
  - We login (using the account we created earlier)
  - We create a new repo, by clicking the green “new” repo button
  - When the repo is created, we get a link.
  - We link this remote repo to our local repo by the following command on your terminal:
    - git remote add origin <your remote repo address”

# Push file to remote repo

- Finally we can push the file to the remote repo using this command:
  - git push
  - We will likely be asked for our password, if using https. If we're using ssh, the keys will be read directly
  - If asked for your password, type it.
  - If we didn't set username and email (during the config step), we may be asked to enter those too
- Now, we can check that our changes are available on the remote repo
  - Go to github.com
  - On the top right, under your username, select repositories
  - This will open your repositories, and the one you just created should be there
  - Click on it to see your "about_myproject.txt" file. You can click the file to open it -- make sure it has what you wrote

# Creating and using git recap

- Download and install git
- Create an account on your version control system of choice  e.g. Github, bitbucket
- Configure your account username and email.
    - You can use ssh here, to avoid typing passwords each time
- Create a local folder and initiate a git repo using git init
- Create your files and add them to your folder.
    - Use *git add* to add files to repo, and *git status* to check repo status
- Commit your changes locally. Use *git commit --m "<your message>"* command
- Create a remote repo on your website of choice
- Connect your local repo with the remote repo using *git add remote origin <remote repo url>* command
- Push changes to remote repo using the *git push* command

# Git Commands

- Git init - initialize a new git repo locally
- Git add - add files/changes to your local repo
- Git commit - commit changes you've made. Remember to add a helpful message
- Git add remote origin - add a remote repo to your local repo
- Git push - push committed changes to your remote repo
- Git pull -- pull changes made from the remote repo
- Git clone -- clone an existing remote repo locally

# Cloning an existing repository

- Sometimes, someone else has done some work and made it available e.g. via Github.
- If they allow you to (based on licence agreement), you can copy the work they've done to run it yourself and/or edit it.
- You would need to clone that repo.
- You first locate where you'd like to copy that repo to on your local machine and navigate there.
    - E.g. cd ~/Document/dev
- You find their git repo url by checking on github for instance
- On your terminal, use the git clone command to get the code from github.
    - E.g. git  clone https://github.com/pmugambi/191-git-tutorial.git
    - General command is git clone <remote_repo_url>

# Assignment 2

- Typically, people use a “README.MD” file to describe their project. If you do so, Github will recognize the file and use it as informational guide to people when they view the repo.
- You are to create a README.MD file for your project and add it to the local and remote repo.
- Additionally, clone my repo here https://github.com/pmugambi/191-git-tutorial.git which has 3 files.
- Copy the “squares.py” file into your “myproject” repo, and add it to your remote repo.
- Submit a document on Moodle containing screenshots of your remote repo, with the readme, and squares.py files.
    - Your username should be clearly visible.

# Recap

- Git is useful for code versioning and backup
- It also allows multiple people to work together on the same code base
- Using features such as branches, many people can work together and combine their changes to one main branch -- usually master
- Git typically can store many forms of files, .docx, .txt, .md, .pdf, .py, .c++, .java etc
    - There is much you can add to your repo
- Github/bitbucket can help you build and showcase a portfolio, which we discussed previously was important for a CS student.
- There are several features I haven't talked about e.g. Ignoring system and other files on git. You'll learn these as you use git more