Intro to Git

IEEE Workshop Committee

Please Sign in!

http://bit.ly/GitHub_SignIn

Install Git

git-scm.com/download

Windows

- Download .exe
- Use default install location
- Use default install settings

- Git Bash

Mac

- Download dmg installer
- Follow instructions

- Terminal

Linux

- Use correct commands for your distro

- Terminal

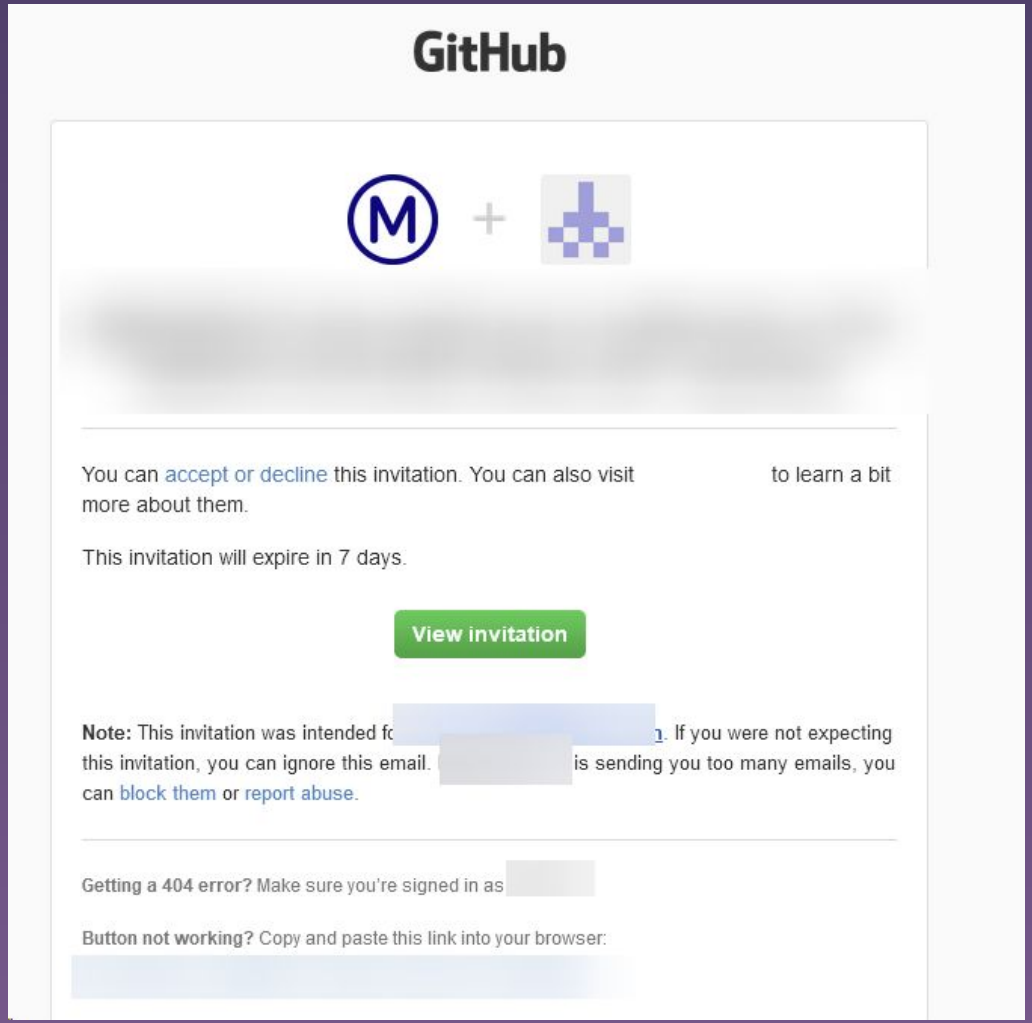
- GitHub Desktop not available

GitHub Desktop Setup

- github.com
  - choose sign-up in top-right
- Write in chat your github username
- desktop.github.com
  - download for MacOS or Windows
  - install, log in, and configure with your info

Paste your GitHub email into Zoom

Look for an email that looks like this:



What is Git?

- Distributed Version Control System
  - Version Control - Records changes to files over time, can revert back to specific versions
  - Distributed - Can “clone” entire repository (current state + previous states), on different computers
- Free, open source
- Lightweight and fast
- Most used VCS

Git vs GitHub vs Gitlab

- Git - Software which handles VC, makes/tracks local changes
- GitHub - Remote hosting of Git repositories, other features like collaboration w/ other users
- GitLab - Similar to GitHub, provides DevOps tools and is open-sourced

Some Core Ideas

- **initialize** - create a new repository
  - folder to be used for a git project
- **add** - adds file to launch pad after it has been changed
  - checkbox next to files in github desktop
- **commit** - bundles together added files
  - adds a message, a checkpoint in the project
- **push** - sends all commits to the remote server
  - updates project with local changes
- **pull** - updates local repo with changes on server repo
  - updates local project with other people's changes
- **clone** - copy server repo to your machine as local repo
  - necessary first step to begin work
- **branch** - allows different things to be worked on
  - different versions being modified, merged when done

Creating & Using a Local Repository

Local Git Setup

- Sign in to GitHub account on GitHub Desktop
- Enter name and email to Configure Git
  - all changes will have these identifiers

Getting Started

Initialize Repository

- Create an empty folder
  - folder will house git repositories
  - My Documents, Desktop, etc.

- File -> New Repository
  - choose new folder
  - name the project, initialize with ReadMe

Stage first commit

- Create a text file
  - Save it in the repo
- Commit that file with a message
  - write a summary and a description
  - click commit
  - change will disappear from list

Make Changes

Stage + Make New Commit

- Open the text file you originally made and make an edit to it
- Repeat the same steps as before to create a new commit
- View your commit history
  - history tab on left side of screen
  - view -> history / ctrl+2

Branches

- Git allows branches
  - Different people work on different branches

- Currently on the main branch, can create and move to another one
  - branch -> new branch

Create a Branch

Making a change on a branch

- Now on a new branch
- Add a new txt file
- Add and commit that file

Merging Branches

- Combine changes on new branch with main
- Switch back to the main branch
  - leave changes behind
- Merge the changes from the new branch
  - branch -> merge into current branch

Using a Cloud Repository

Get Access to our Repository

- Go to email associated with your github account
- Find the email and accept the invite
- Go to the repo on github.com

Clone the GitHub Repository

- A linked copy of the GitHub repository can be created locally

- file -> clone
- select the repository, choose a folder for the repo to sit in
- If not there, log out of GitHub Desktop and log back in

Create a Branch + Make Changes

- Now that we've cloned the cloud repo, we can contribute changes to it
- Create a new branch with your Drexel ID as the name
  - Not good practice to commit directly to main

- Add a new file and stage the commit
- Push the changes back up to GitHub
  - make sure you're on your branch, not main

GitHub Pull Requests

- On remote repositories, want to talk about stuff before merging changes
- Pull Requests allow collaborators to discuss the merge
  - Even make some necessary commits beforehand
- Pull requests end with a branch being merged into another and often being deleted

Pull from Repository

- Switch back to the main branch
  - Note that no changes are present, like nothing happened

- Choose pull
  - Get the new file to your local repo from the cloud

Additional Topics

- .gitignore - Specifies files which are to be ignored by git
  - Can provide specific file or give a pattern
  - Used in large-scale software projects
  - If using an IDE, use .gitignore
- Using bash commands (Git Bash)
- Rather than doing a hard reset, you can temporarily revert to a previous commit on a new branch

Raffle

Questions?

http://bit.ly/IEEE_GIT_FEEDBACK