ECEN 3233: USING A GIT REPO

Nate Lannan

Assistant Professor of Practice

Oklahoma State University

Electrical and Computer Engineering Department

203 General Academic Building

(405)744-8040

nate.lannan@okstate.edu

GitHub As A Repo

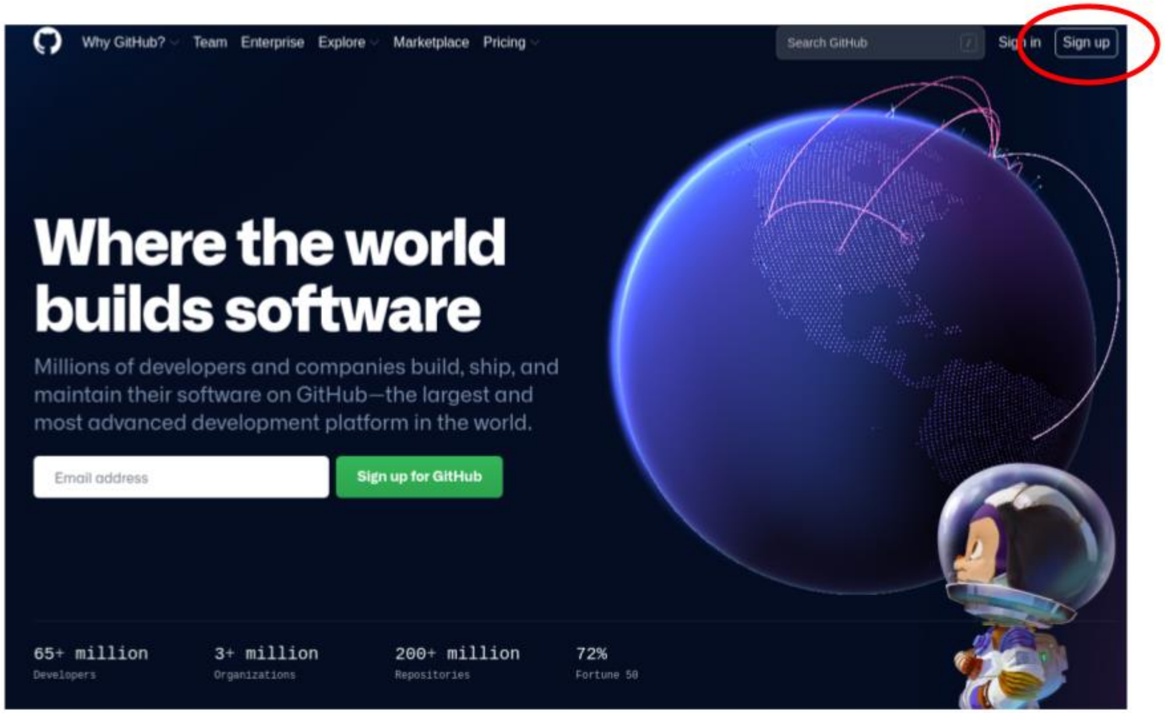
- Up until this point we have been using GitHub as a glorified cloud storage system (simple clone of github repo)
  - Some of the more experienced users have forked the repo
  - Please refrain from committing code to these forks as any of your fellow students can see your code and copy it

- The following slides are the steps to set up your own private repos for your labs
- Why would we bother with this instead of just saving our progress to a thumb drive?

1. Version control – you can revert to a previous save point
2. Both teammates have simultaneous access to your lab files so that you can both work from home at the same time
3. This is what is done in industry and academia

Create a GitHub Account

1. Navigate to https://github.com
2. Select sign up
3. I recommend using your okstate.edu email account so that you can sign up for a student account for more benefits



Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

65+ million Developers

3+ million Organizations

200+ million Repositories

72% Fortune 50

Create your own repo from our templates

- Navigate to the 3233 lab template page for the lab you wish to work on
  - Lab0: https://github.com/natelannan-osu/dldLab0_Spring23
  - Lab1: https://github.com/natelannan-osu/dldLab1_Spring23
  - Lab2: https://github.com/natelannan-osu/dldLab2_Spring23
  - Lab3: https://github.com/natelannan-osu/dldLab3_Spring23
- Select "use this template" -> "create a new repository"
- Create a name for the repository and make it "private"

Setting up SSH Keys

- Turn on Open SSH on windows
  - Open "Services" from start menu
  - Find OpenSSH Authentication Agent -> right click -> properties
  - Change startup type to "Automatic (Delayed Start)"
- Open cmd prompt (I recommend terminal)
  - Start the ssh agent by typing
    ssh-agent
  - Create keys by typing
    ssh-keygen -t ed25519 -C "your github email address"
  - Follow the prompts to create keys. Use the default location and create a password for the keys if you wish
  - Keys will be stored at C:\Users\<your user name>\.ssh\

Adding keys to ssh-agent and github

- In Terminal type
  ssh-add C:\Users\<your user name>\.ssh\id_ed25519
- Enter your password if prompted
- Copy the public key to the clipboard by typing
  clip < C:\Users\<your user name>\.ssh\id_ed25519.pub
- Navigate to https://github.com -> settings from dropdown in upper right
- Navigate to ssh and gpg keys and select "New SSH Key"
- Create a name for the key and paste the key from the clipboard to the key field
- Select "add key"

Clone your repo

- Navigate back to your repo home page and select Code
- Select SSH under Clone
- Copy the git address. in my case: git@github.com:natelannan/myLab0.git
- Go back to the cmd prompt and type
  git clone git@github.com:natelannan/myLab0.git
- Enter your password if prompted

- You now have a working copy of the repo you created. 5 step process for using your repo:

1. Every time you start working first type:
   git pull
2. Make your changes to files and save the changes using whatever text editor you want to use
3. Add your changes to your local repo by typing
   git add *
4. Commit your changes to your local repo by typing
   git commit –m "some meaningful message about the changes you made"
5. Commit your changes to the remote repo on github.com so that your partner can access them by typing:
   git push