GitLab Zoom Notes

* Ls = list files in the current folder
* Cd = change to a different directory
* Mkdir—make a new directory
* Start the line with ‘’fix”, “add”, or “change
* If you need to revert back to old code you can identify which commit you want to go to
* Potential employers will check your Github
* Companies will expect you to write good messages

Tasks

* Create new repository

Branches

* Let you create features independent of the main codebase
* Useful for working on separate features
* **Check status often**
* Branches are expendable
  + Easily deleted, little issue if deleted
* -d your-branch-name – delete branch
* -b your-branch-name -- create new branch

e.g.

* friend\_name = “griffen”
* print(“hello ”+friend\_name)
* Git makes merging easy
* Conflicts – wehen 2 collaborator change the same file in some way, creating 2 versions of the file
* When this happens git doesn’t know which is correct, so the developer has to resolve the conflict
  + Solution is often not binary—often a combo of certain changes from both
  + Often git will open up the merge conflict in a GUI or text editor
* Remote Repositories
  + Github allows you to upload/store your repository
  + <http://qscu.org> – view workshop
  + Git push—
* In groups
  + Choose one person to be owner of github repository
    - He/she adds collaborators
  + Each person should clone the repository to their computer
  + Each person should create a branch
  + Each person should add a file with their name to their branch
  + And each person should push their branch
* Workflow Tips and Tricks
  + README—helpful for providing info on your project to others
  + .gitignore—choose which files not to track
    - Good for security
  + License—useful for not getting sued
    - MIT license is recommended
  + Adding collaborators
  + Then clone the repository into the local system
  + Branches should be created for new features
    - Avoid conflicts with collaborators
  + Add good descriptions
  + ‘Undo’ command—git revert <commit>
  + Git reset <commit>
  + Git stash—temporarily stores your changes so you can switch branches and perform other operations
  + Git stash pop
  + Git stash clear
  + Git diff—to see changes