Version Control

Using GIT to manage projects including tips and tricks; including pycharm/intelliJ GIT integration

Git Basics

- **Working with the command line.**
  - Cloning vs Downloading the zip
  - Init
  - Checkout, Commit, Push, Pull, Merge
  - Stash… pop apply, drop
- **Local Repository or Remote**
  - Benefits?
- **Git Ignore**

Things to Do

- **Adding Projects to remote sources**
- **Having Multiple Collaborators**
- **Public and private repos**
  - Github for open source
  - Hosting your own repository(GitLab)
- **Or for solo projects, easily share between multiple computers**

Using a graphical Git Client

- **Popular Examples of git clients:**
  - SourceTree, GitKraken, Github Desktop
  - Plus many obscure and UI dated ones…
- **Reasons to use standalone git client**

  - Visual see branches and commits

  - No commands to remember

  - Easier to bring coworkers up to speed

Branching And Merging

- **Not a lesson in How to, but why.**
- **Able to work and commit new features, without hurting other developers**
- **Sometimes only git maintainers will handle merging**
  - Merge Conflicts. Which ones to keep
- **Pull Requests**

IDE Integration

- **Many have built in GIT (and older version control) tools built into them.**
- **Local change history, Checkout from VCS, Import to Repo, or Init a repo. All from your IDE.**

PyCharm Example

Can’t find software for your OS?

- Check the github!

  Often organizations will put source or hide unofficial executables on Github.
- Lazy Developers don’t want to compile for every architecture.

Git command tips

- **git help -g**
- **External merge tools. (Why all or nothing?)**
  - Gitmerge for unity
- **Stash your local changes**

Alternate uses for Git

- Saving .config files between machines
- Too lazy to download google drive/dropbox?
- Not just for programmers, git can be used for other professions that share files.

Questions, comments, thoughts?