A Gentle Introduction to Git

North Austin Pythonistas
December 2019

Goals

* What is git?
* Benefits of source code control.
* How to setup a local git repository.
* How to connect your local repo to GitHub.
* Learn some simple development workflows.

What is git?

Distributed file version control

1. Records changes to a file.
2. Helps manage multiple edits to files.
3. Works for any kind of file!
4. Supports team collaboration.

Benefits of Source Code Control

* Revert changes made to files easily.
* Tools for reviewing file differences.
* Commit logs indicate who did what and why.
* Helps teams easily merge their changes.
* Distributed source is harder to lose.

Setup a Local git Repository

* Make a new directory and initialize it:

1. $ mkdir ~/local/new-project
2. $ cd ~/local/new-project
3. $ git init
4. $ ls -l
5. .git

Definition: What is a git Repository?

* A directory with a .git hidden directory
* Files managed by git in:
  1. The current directory
  2. All non-empty subdirectories.
* Generally shortened to “repo”

Connect a Local Repository to Github

1. Create a repo on GitHub
2. Create a repo locally on your machine
3. Synchronize your repo with GitHub

Creating a GitHub Repository

1. Create a GitHub account @ https://github.com
2. Find the “+” drop down menu on the right
3. Select “New Repository”
4. Fill out the form, name is all you need.
5. Done!

Some Git Housekeeping

* Set your email and username

```
$ git config --global user.email yourname@example.com

$ git config --global user.name "Your Name"
```

I *generally* use my GitHub ID for user.name

Sync’ing a Local Repository with GitHub

* This is voodoo, sorry.

```
git remote add origin https://github.com/UserName/RepoName.git

git push -u origin master
```

Good news!

When you create a new repo on GitHub, these directions are repeated there so you don’t have to look these up.

A Simple Git Workflow

* Changing a file and updating remote repo:
  1. git status
  2. git add path/to/file
  3. git commit
  4. git push

A Simple Git Workflow – Reverting Changes

* Before an add:

1. vi filename
2. git checkout -- filename

A Simple Git Workflow – Reverting Changes

* After add and before commit:

1. vi filename
2. git add filename
3. git reset HEAD filename
4. git checkout – filename

Git Resources

* **Software**

  https://git-scm.com/downloads

* **Documentation**

  https://git-scm.com/doc

  https://git-scm.com/book/en/v2

  https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf

* **How To Fix Things When They Inevitably Get Weird**

  https://github.com/k88hudson/git-flight-rules

* **Make Your First GitHub Contribution!**

  https://github.com/firstcontributions/first-contributions