An introduction to Git and GitHub

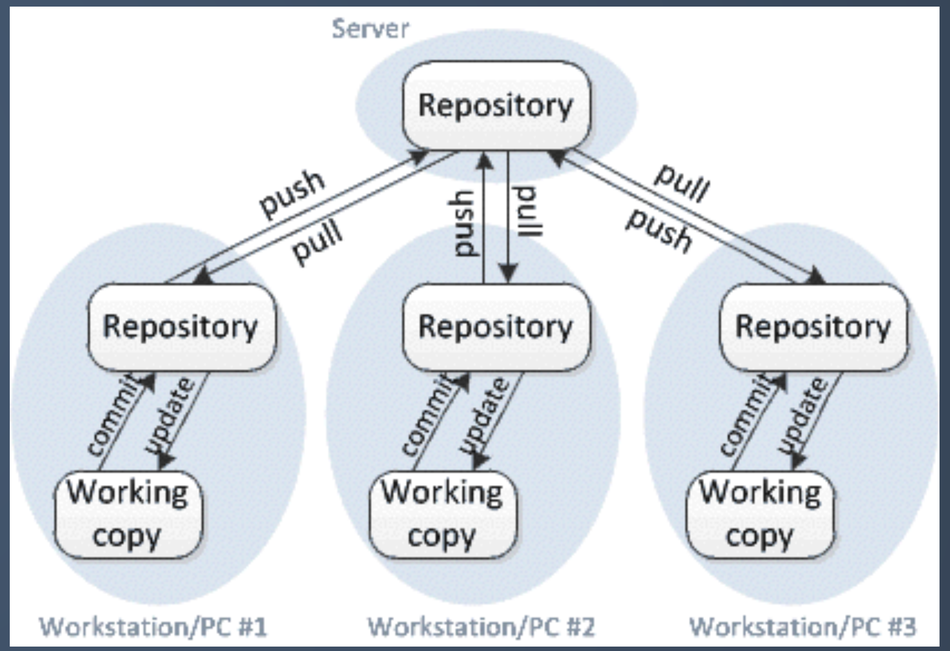
Knoxville CocoaHeads - October 2015

What is Git?

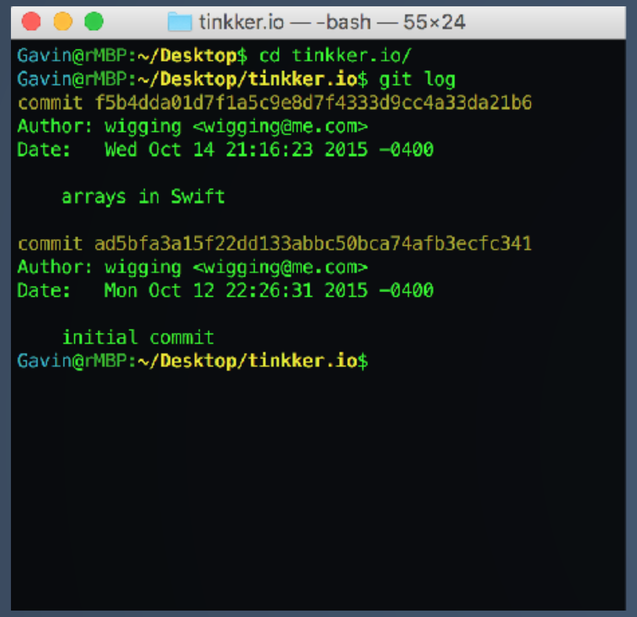
- Free and open-source distributed version control system
- Created in 2005 by Linus Torvalds for development of the Linux kernel
- Already installed on Macs as part of OS X
- More info at https://git-scm.com

Distributed version control

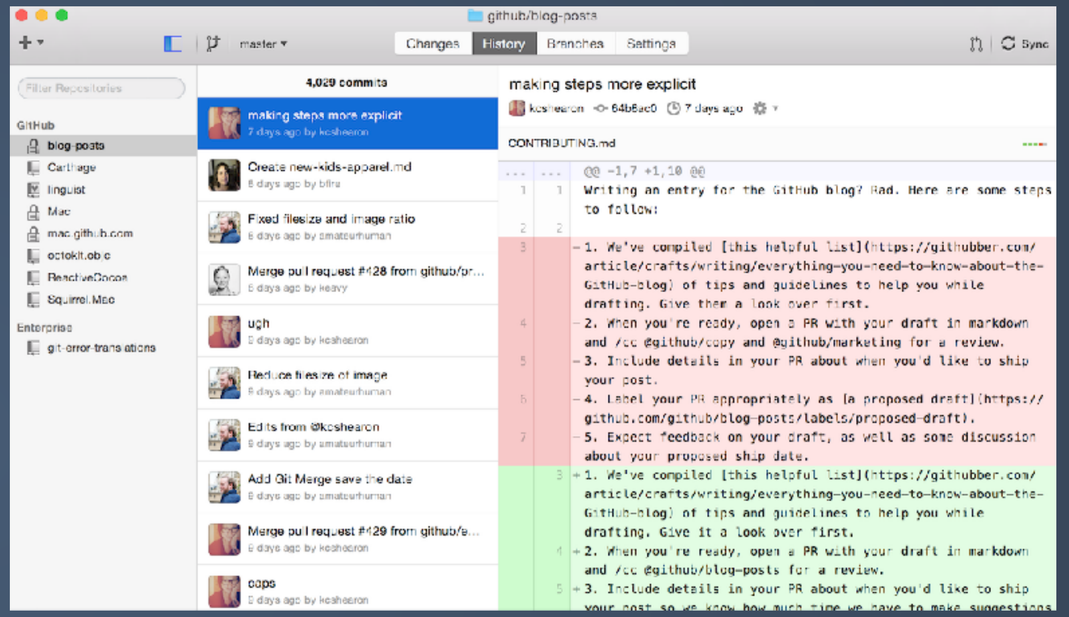
- Version control is a system to record changes to files over time so that you can recall specific versions later
- Git is a Distributed Version Control (DVC) system where each repository is fully mirrored
- Each "clone" of a repository is the full history of a project



Command line or GUI



Command line interface



Desktop apps

https://desktop.github.com/

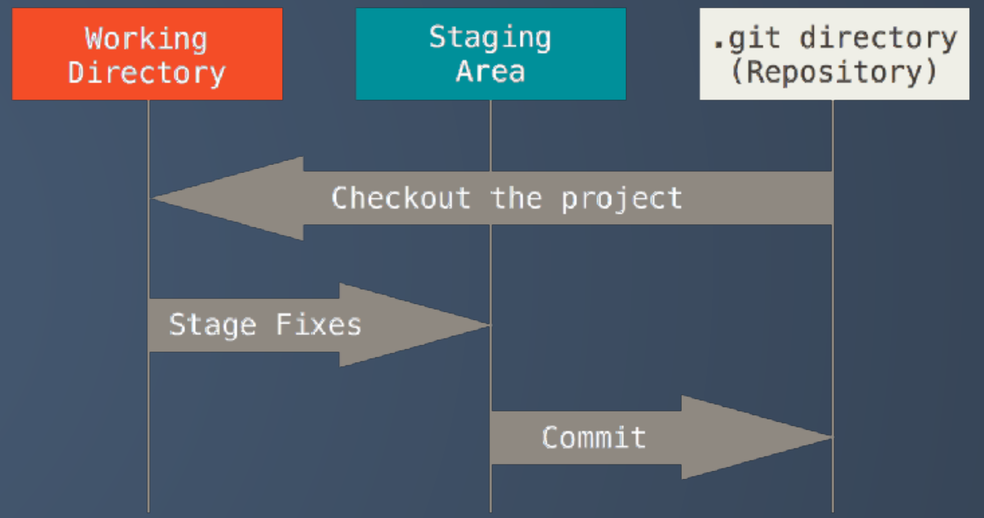
http://www.git-tower.com/

Main sections of a Git project

Working directory - single checkout of a version of the project, the files on disk to use or modify

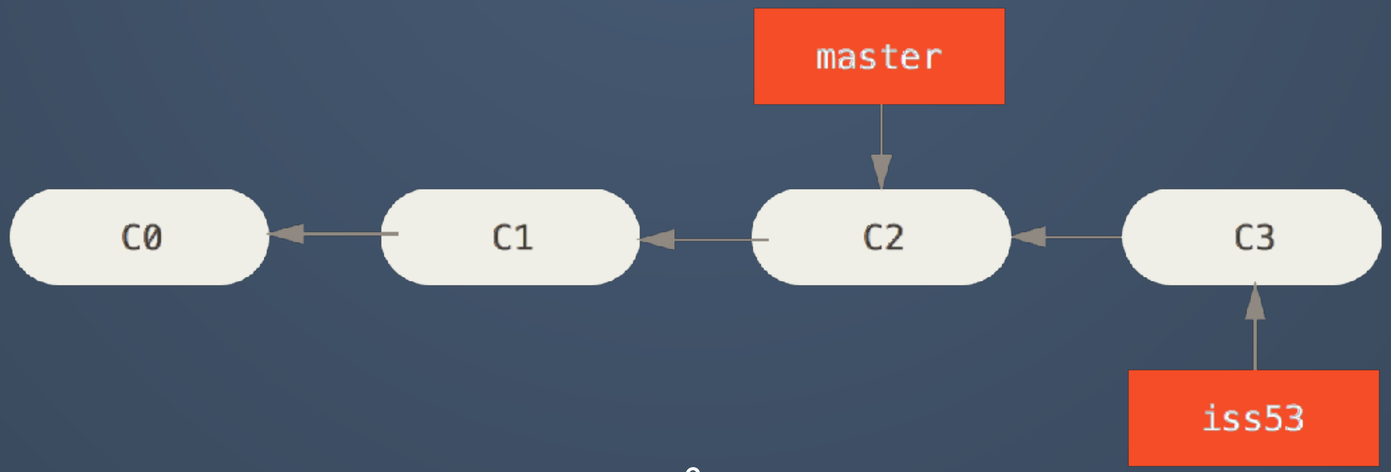
Staging area - information of what will be committed to the database

Git directory - the metadata and object database for the project, what is copied when cloning a repository from another computer



Branching in Git

- A branch allows the user to work on a separate development path without affecting the main work flow
- Useful for beta features
- Main branch is typical the "master" branch
- Create a new branch with the git branch command
- Use git checkout to switch to an existing branch
- Merge development from one branch to another with the git merge command



Configure your Git environment

- Use the git config command to customize Git on your local machine
- Username and email associated with commit messages
- Define a text editor for commit and tag messages

```
$ git config --global user.name "Johnny Appleseed"
$ git config --global user.email johnny@mail.com
$ git config --global core.editor vim
```

Initialize the Git repository

- Create a new Git repository with the git init command
- The .git subdirectory contains the repo files

```
$ git init
Initialized empty Git repository in /Users/Gavin/Desktop/zGit/.git/
```

Basic Git workflow

- Modify and/or add files in the working directory
- Add the files and modifications to the staging area
- Commit files from staging area to store them in the Git directory as a permanent snapshot
- Show the current status of the project with *git status*
- Show a list of commit logs with *git log*

```
$ git add .
$ git status
On branch master
Initial commit
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   fileA.txt
$ git commit -m 'initial commit'
[master (root-commit) b2d2928] initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 fileA.txt
$ git status
On branch master
nothing to commit, working directory clean
```

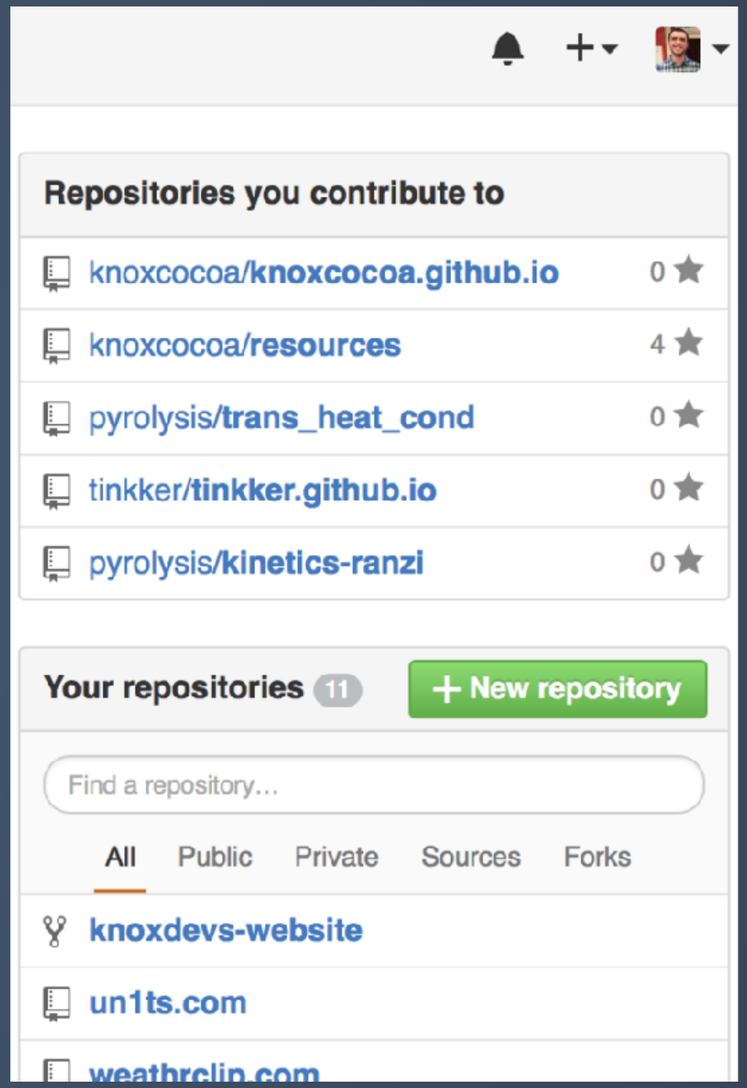
Demo…

What is GitHub?

- A hosting service for Git repositories
- Provides issue tracking, code review, wiki, etc.
- Offers other services such as
- Gist for code snippets
- Speaker Deck for presentation slides
- GitHub Pages for website hosting
- More info at https://github.com and https://git-scm.com

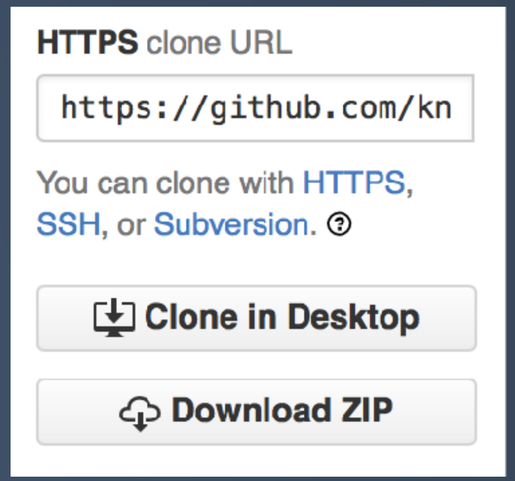
Setup a GitHub repository

- Create a new repository on GitHub with the "New repository" button in the dashboard or the "+" button in the toolbar
- Repositories on GitHub can be public (free) or private (paid)
- Every repository is accessible via https://github.com/user/reponame
- Each repo should contain a README.md to describe the contents of the project, markdown is render as HTML on GitHub
- GitHub repositories can also host websites with custom domain names, see GitHub Pages



Cloning a project on GitHub

- The git clone command will copy a repository to a new directory (downloads the project from GitHub)
- Tracks remote branches associated with cloned project
- Use git fetch or git pull to update from remote branch
- The git push command will update the remote repository on GitHub with changes made to the local cloned files




```
$ git clone https://github.com/pyrolysis/literature.git
Cloning into 'literature'...
remote: Counting objects: 185, done.
remote: Total 185 (delta 0), reused 0 (delta 0), pack-reused 185
Receiving objects: 100% (185/185), 100.17 KiB | 0 bytes/s, done.
Resolving deltas: 100% (73/73), done.
Checking connectivity... done.
```

Demo…