Git Lecture Material – Jake Sorce / Dave Jungst

DEVPOINT LABS

What is Git?

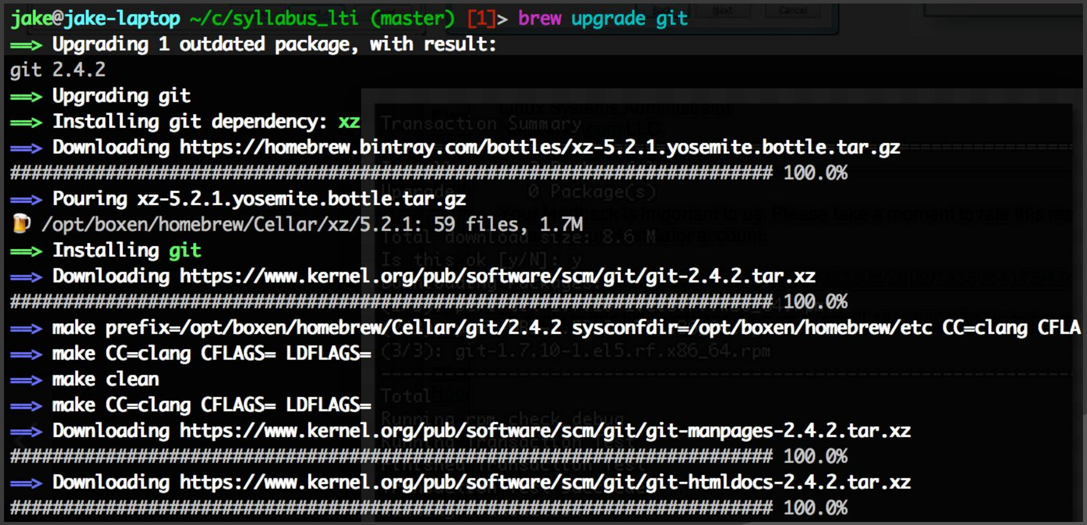
- Technical Term: Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows
- Layman's terms: A secure place to store your project files and easily collaborate on them with others.

Where did Git come from?

- Developed by Linus Torvalds and the Linux Development Community
- Developed with lessons learned from using BitKeeper
- Goals:
  - Speed
  - Simple
  - Non-linear development support (parallel branches)
  - Fully distributed
  - Large project support

Installing Git

- http://git-scm.com/download/mac
- brew install git



```
jake@jake-laptop ~/c/syllabus_lti (master) [1]> brew upgrade git
==> Upgrading 1 outdated package, with result:
git 2.4.2
==> Upgrading git
==> Installing git dependency: xz
==> Downloading https://homebrew.bintray.com/bottles/xz-5.2.1.yosemite.bottle.tar.gz
######################################################################## 100.0%
==> Pouring xz-5.2.1.yosemite.bottle.tar.gz
🍺  /opt/boxen/homebrew/Cellar/xz/5.2.1: 59 files, 1.7M
==> Installing git
==> Downloading https://www.kernel.org/pub/software/scm/git/git-2.4.2.tar.xz
######################################################################## 100.0%
==> make prefix=/opt/boxen/homebrew/Cellar/git/2.4.2 sysconfdir=/opt/boxen/homebrew/etc CC=clang CFLA
==> make CC=clang CFLAGS= LDFLAGS=
==> make clean
==> make CC=clang CFLAGS= LDFLAGS=
==> Downloading https://www.kernel.org/pub/software/scm/git/git-manpages-2.4.2.tar.xz
######################################################################## 100.0%
==> Downloading https://www.kernel.org/pub/software/scm/git/git-htmldocs-2.4.2.tar.xz
######################################################################## 100.0%
```

Generating SSH Keys

- ssh-keygen -t rsa -b 4096 –C your_email@example.com

- Passphrase is strongly recommended but not required

Github

- http://www.github.com

What you should expect to learn

- Basic Git configuration
- Creating a new repository
- Status and Log
- Staging and Committing files
- Branching and Merging
- Stashing
- Creating a new repository - Github
- Cloning an existing repository - Github
- Pushing to a repository – Github
- Pulling from a repository - Github
- Amending a Commit

Git Basic Configuration

- git config --global user.name "John Doe”
- git config --global user.email johndoe@example.com
- git config --global core.editor vim
- .gitignore – a file in the git repo to determine what git should ignore

Creating a repository

Status & Log

- git status – shows current status of repo
- git log – shows history of commits to the repo

Staging Files

- Staging – A loading dock where you get to determine what changes get shipped away.

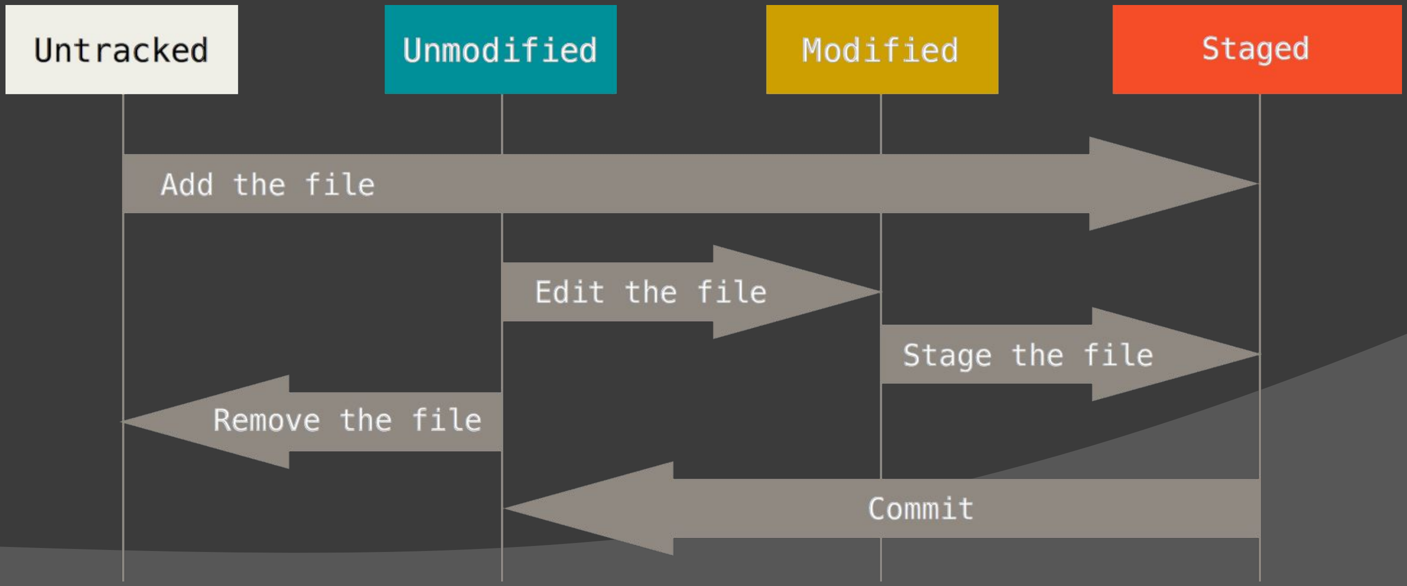
- git add filename.html
  - This command will stage the file to be committed into the repo

Committing Files

- Committing - commit snapshots of your changes into your repository each time the project reaches a state you want to record.

- git commit -m ‘my first commit’

- git commit –amend
  - This will put staged changes on top of the previous commit.



Un-Staging Commits

- git reset HEAD <file>
  - unstages the file
    - HEAD is tip of the branch

- git reset — hard
  - blows all changes away

- git checkout – <file>
  - reverts file back to what was previously committed

Branching

- git branch <branch_name>
  - This creates a new branch

- git checkout <branch_name> – puts you on that branch to do work on

- git checkout -b <branch_name>
  - creates a new branch and switches you to the branch

Merging Branches

- git checkout master
  - put yourself back on the master branch
- git merge <branch_name>
  - merge all changes in the branch into the master branch

Branches cont…

- git branch
  - Lists all checked out branches
- git branch –v
  - Lists all checked out branches and the last commit on that branch
- Git branch –d <branch>
  - This will delete a branch (*note: you cannot delete a branch you are on!*)

New Repository - Github

- https://github.com/new

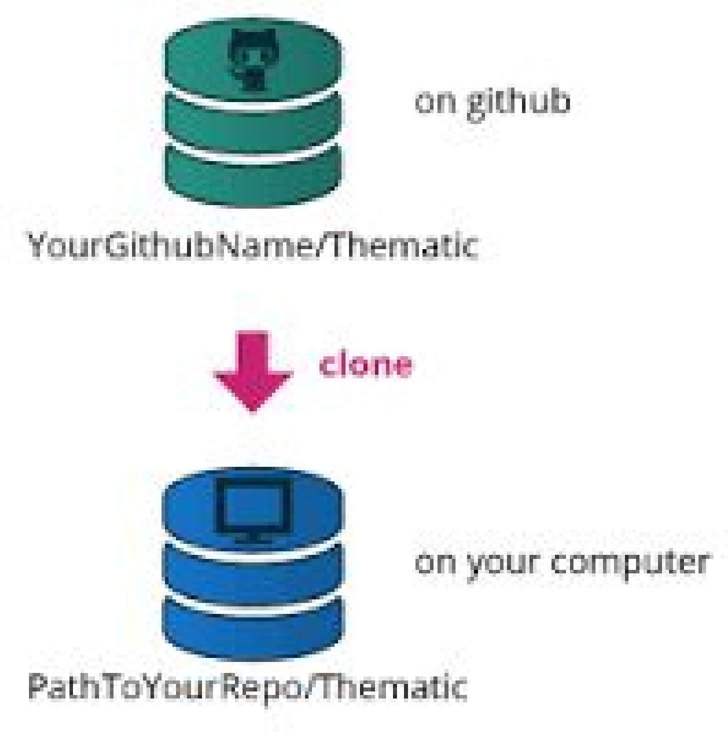
Adding remotes to existing project

- git remote –v
  - Lists all remotes for current repository

- git remote add <remote name> <url>
  - Adds a remote
    - origin is the default remote to be pulled and pushed and should be added as your main remote

- git remote remove <remote name>
  - Removes a remote

Clone Repository - Github



Pushing to a Repository - Github

Pulling from a Repository - Github

- git pull <remote name> <branch name>
  - Grabs online updates and merges them with your local work

Mini Project

Create a new repo for cheatsheet

Push cheat sheet to github

Follow this link: https://classroom.github.com/assignment-invitations/62583ba8ae62d53951237799de89a34f

This will create a repository for you to submit your assignment