

# INTRO TO GIT/GITHUB

AN OXY OPEN SOURCE WORKSHOP



# Welcome!

## Today's Agenda

What's the point?

Git vs. Github

Version Control and its terminology

Using Git/Github

# Before we get started

---

First you will need to create a Github account (use your oxy email for the free perks!!): <https://education.github.com/pack>

Then, make sure you have Git on your computer:

MAC OS
You're good

WINDOWS
Visit <a href="https://git-scm.com/">https://git-scm.com/</a> And download Git

Finally, join our Github Organization: <https://github.com/oxy-opensource>

# Did you know...

---

~25,000,000

People use Git or Github

“According to web traffic monitor Alexa, GitHub is now among the top 100 most popular sites on earth.” -Wired

Blender, R, Atom, Git, and so many advanced technology are byproducts of open source.

*It is hard to finish projects on your own.*

# What's the point? (cont.)

---

Internships, jobs, projects (tech resume)

Free services for students

Have your code reviewed by the open source community (and vice versa)

Collaborate projects with the local/global community

Track changes of code

Contribute back to the open source community

# What is Git?

---

Git is a **version-control system** that holds the entire history of your code. (savepoints = commit)

Git gives you the freedom to **pursue new features & ideas** without ruining your code.

Git fundamentally uses **command line**

# Basic Command Line

---

`cd [directory name]` - change directory

`cd ..` - go one directory up

`pwd` - get current path

`ls` - list files of current directory

`mkdir [dirName]` - makes a new directory (file)

`touch [fileName.fileType]` - creates a file

`rm [fileName]` - removes a file

# Command Line

---

Git uses `command line`, which can be intimidating for many people. Github makes interacting with the Git technology easier (but we will be using `command line` for practice!)



# What is Github?

---

Github is a **Git repository hosting system** that puts Git technology on a web graphical interface

Github is a “hub” where users can publicly/privately store their Git repos for **collaboration** from the community

Encourages the open source movement!

# Features of Github

## Discover open source projects

The screenshot displays the Google Open Source website. At the top, the Google logo is followed by the text "Google" and the URL "https://opensource.google.com/". Below this, there are two tabs: "Repositories 1,099" and "People 1,560". A search bar labeled "Search repositories..." is present, along with filters for "Type: All" and "Language: All".

The main content area lists several open source projects, each with a title, description, and statistics. The projects shown are:

- googet**: GooGet package manager. Statistics: 22 stars, 10 forks, Apache-2.0 license, updated 2 minutes ago.
- neuroglancer**: WebGL-based viewer for volumetric data. Statistics: 124 stars, 51 forks, Apache-2.0 license, updated 2 minutes ago.
- skia**: Mirrored from <https://skia.googlesource.com/skia>. Description: Skia is a complete 2D graphic library for drawing Text, Geometries, and Images. Statistics: 1,725 stars, 420 forks, C++ language, updated 7 minutes ago.
- boringssl**: Mirror of BoringSSL. Statistics: 258 stars, 111 forks, C language, updated 7 minutes ago.
- google-auth-library-java**: Open source Auth client library for Java.

On the right side of the page, there are two sections:

- Top languages**: A list of languages with corresponding colored dots: Python (blue), Java (orange), C++ (pink), Go (dark blue), and JavaScript (yellow).
- Most used topics**: A list of topics in blue boxes: android, python, go, java, and google.

At the bottom right, there is a section titled **People** with a grid of 12 profile pictures and the text "1,560 >". Below this, it says "Developer Program Member".

Two annotations are present on the image:

- An arrow pointing to the "googet" project with the text "Description of the project".
- An arrow pointing to the "skia" project with the text "Popular language for the project".

# Features of Github

## Solve/create issues of projects

The screenshot displays the GitHub web interface for the repository `oxy-opensource / oxy-opensource.github.io`. The top navigation bar includes links for `<> Code`, **Issues 4**, `Pull requests 0`, `Projects 0`, `Wiki`, `Insights`, and `Settings`. On the right, there are buttons for `Unwatch` (2), `Unstar` (1), and `Fork` (0).

A central message box states: "Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with `help wanted` or `good first issue`." A `Dismiss` link and a `Go to Labels` button are present.

Below this, a search bar shows the filter `is:issue is:open`. To the right are tabs for `Labels` and `Milestones`, and a green `New issue` button.


The issues list shows 4 open issues and 0 closed issues. The issues are:




- `Design a new club logo` (non-code) #4 opened 2 days ago by emilyttran
- `Add logo` (enhancement) #3 opened 2 days ago by emilyttran
- `Update officer roles` (update) #2 opened 2 days ago by emilyttran
- `Establish color theme for club` (enhancement) #1 opened 2 days ago by emilyttran

Note: You do not have to code to contribute!

# Features of Github


Look at commits (savepoints)/collaborate




 [angulostephanie](#) / [StephaniesAndEmilysRepo](#) Private


 Watch 0  Star 0  Fork 0




[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#)


History for [StephaniesAndEmilysRepo](#) / [about.txt](#)




 Commits on Oct 29, 2017




 **Emily added a cool text art**  
emilyttran committed 10 days ago  8947689 




 Commits on Oct 27, 2017

 **Emily added her response**  
emilyttran committed 11 days ago  32ec306 

 Commits on Oct 25, 2017

 **Added a photo to readme and response to txt file**  
angulostephanie committed 13 days ago  75d7440 

 **Added mi bio**  
emilyttran committed 13 days ago  94affba 

 **First commit, hi emily! :)**  
angulostephanie committed 13 days ago  583deaa 

# Features of Github

## Look at specific changes of code

```
✦ @@ -10,4 +10,29 @@ Stephanie's response:
10 10 I have a love hate relationship with vim lmao.
11 11
12 12 Emily's response to Stephanie's response:
13 -Same though. I figured out to avoid it :D Use start [filename] because open isn't a command for my bash for some reason :D
13 +Same though. I figured out to avoid it :D Use start [filename] because open isn't a command for my bash for some reason :D
14 +
15 +Emily:
16 +LOOK
17 +-----
18 +-----
19 +-----
20 +-----
21 +-----
22 +-----
23 +-----
24 +-----
25 +-----
26 +-----
27 +-----
28 +-----
29 +-----
30 +-----
31 +-----
32 +-----
33 +-----
34 +-----
35 +-----
36 +-----
37 +-----
38 +-----
```

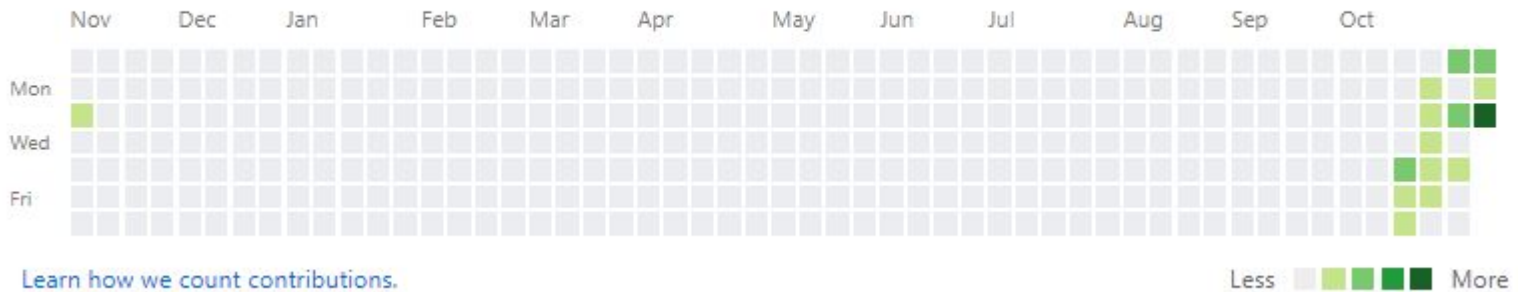
Red = Deletion | Green = Addition

# Features of Github

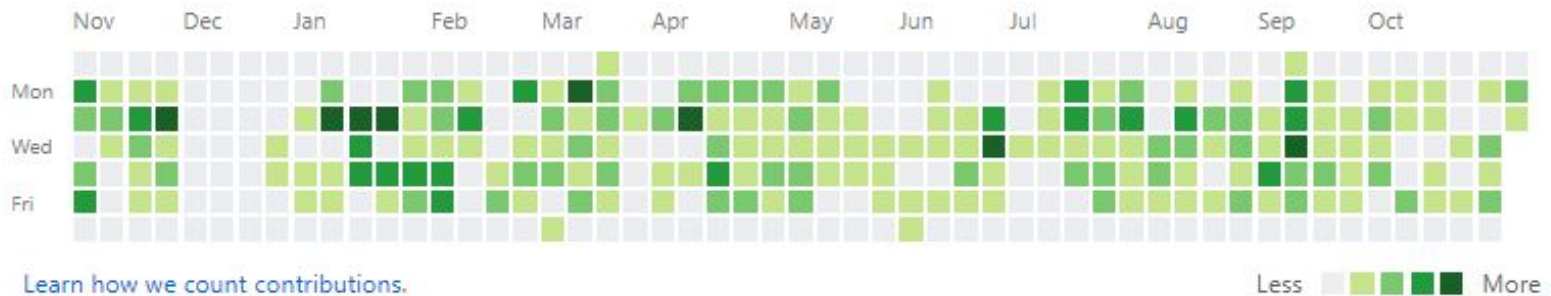
## Track # of your contributions

53 contributions in the last year

Contribution settings ▾



969 contributions in the last year



Hopefully you can see why this is an important feature for recruiters

You  
vs  
The guy she tells you  
not to worry about

# Review: Git vs. Github

# Git

The version control technology that uses command line

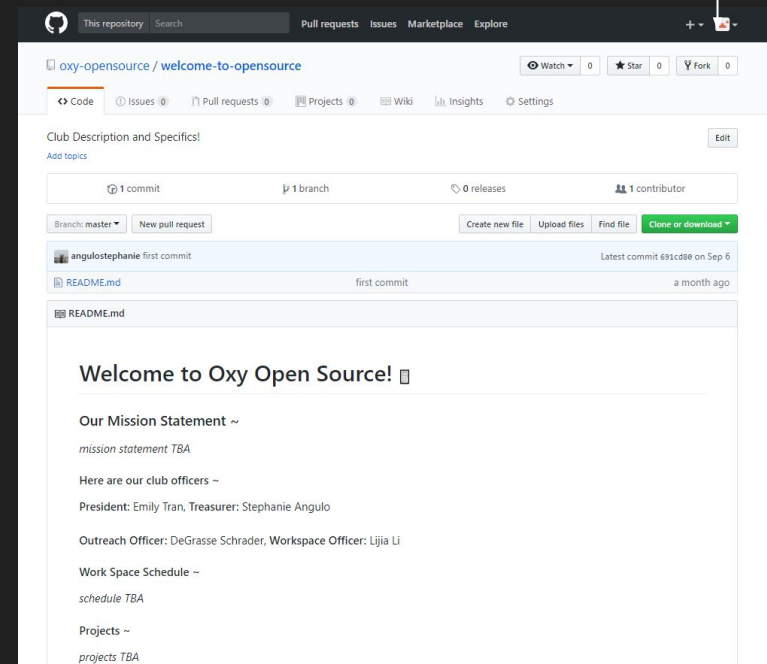
```

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Committer: unknown <Jason@Jason-Quad-PC.(none)>
#
# On branch master
#
# Initial commit:
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       three new files to be ignored:
#       new file:   .gitignore
#       new file:   .htaccess
#       users of new file:   HTMLTable.class.php
#       new file:   My Date.class.php
#       gitignore of new file:   MyLocalization.class.php
#       new file:   Sharp UMS / DS Store
#       match your new file:   Sharp UMS/docs/controllers/htmlred
#       new file:   Sharp UMS/docs/emailtemplates.html
#       new file:   Sharp UMS/docs/index.html
#       new file:   Sharp UMS/docs/installation.html
#       you're done new file:   Sharp UMS/docs/models.html
#       new file:   Sharp UMS/docs/mvc.html
#       your code new file:   Sharp UMS/docs/overview.html
#       new file:   Sharp UMS/docs/resources/img/background.jpg
#       git DIR/new file:   Sharp UMS/docs/resources/img/bg.png
e:\Server\htdocs\myframework\git\COMMIT EDITMSG [unix] (06:40 04/12/2011) 1,0-1 Top
git - not properties found on file: e:\myproject\gitignore.txt

```

# Github

Web application  
that uses Git  
and interacts  
with users



**Git ready ...**



# Some Important Terminology

---

**Directory** - folder

**Repository (“repo”)** - collection, project

**Commit** - savepoints

**Local vs. Remote** -

Local
Refers to your computer server

Remote
Refers to an external server (e.g. Github)

# Getting Started

---

You will need to connect Git to your Github account.  
In your command line, please

Open your terminal (MAC/Linux) or Open Git (Windows)

Type `git config --global user.name "[First Last]"` (ie. "Mona Lisa")

Type `git config --global user.email [YOUR EMAIL]` (ie. `oswald@oxy.edu`)

Check to see if this was done correctly!

Type `git config --list`

# Songwriting (Demo)

---

Say you want to need to write lyrics for a song.  
Let's create an about txt file to start that process.

Make a directory (aka a folder) for your project!

Type `mkdir AwesomeSong`

Go into your newly created directory

Type `cd AwesomeSong`

Check to see if you have successfully changed directories

Type `pwd`

*This should return the full path name, with your new directory at the end.*

*Example: /Users/stephanieangulo/Desktop/AwesomeSong*

Create a txt file in your repo/project folder

Type `touch lyrics.txt`

Check to see if file was successfully created

Type `ls`

# Songwriting (Demo)

---

Let's add a song title to your project

Open the txt file

Type `open lyrics.txt` (MAC)

Type `start lyrics.txt` (Windows)

Input a song title in lyrics.txt

Type [song title] (ie. *"I'm in Love with a Coder"*)

Save the txt file through the txt program

# Songwriting (Demo)

---

Now that you made a song title, let's create a savepoint.

Now initialize a git object for your project

Type `git init`

Check to see if the git object was successfully created

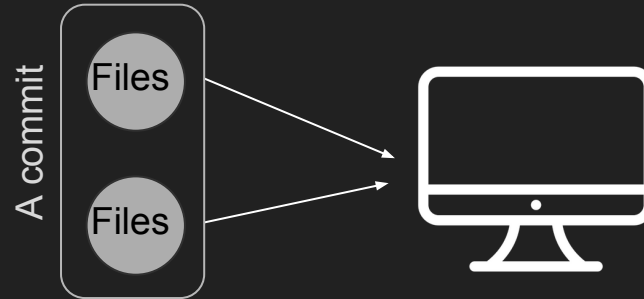
Type `ls -a`

*(there should be a ".git")*

# Workflow (Demo)

---

You will want to create (“commit”) your file(s) to your local git object



Add the file(s) for preparation (“staging” files for commit)

Type `git add lyrics.txt` or `git add .`

Check to see if git successfully added the file for preparation

Type `git status`

Commit the added files and include information about this upload

Type `git commit -m “initial commit”`

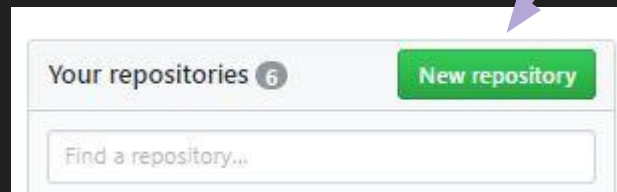
# Workflow (Demo)

---

Now, you will need to create a repo for your app project to hold these savepoints/commits. We will create it remotely on Github, and then connect it to our local machine



Login to your Github and create a New Repository called "Awesome-Song"



# Workflow (Demo)

---

Since the repo is remotely on Github, you will need to connect the remote repo to your local server (aka your computer)



Go on Github and copy the HTTPS URL for your repo

Connect the remote repo to your local repo

Type `git remote add origin [HTTPS URL] + ENTER`

Check if connection was successful

Type `git remote -v`



# Songwriting (Demo)

---

OH NO! It turns out that none of your friends like your song title. Let's change the song title and commit that change (with commit message "changed song title")

## Commands:

```
git add  
git commit -m "[message]"  
git status
```

## Challenge:

Upload an img to your AwesomeSong directory (you don't have to use command line for this) and commit this file with message "added song photo"

# Songwriting (Demo)

---

You can continue committing with other files (e.g. code, more txt, images). Every commit is local. In order to update your repo remotely on Github so that the public can see your repo, you will need to push your commits.



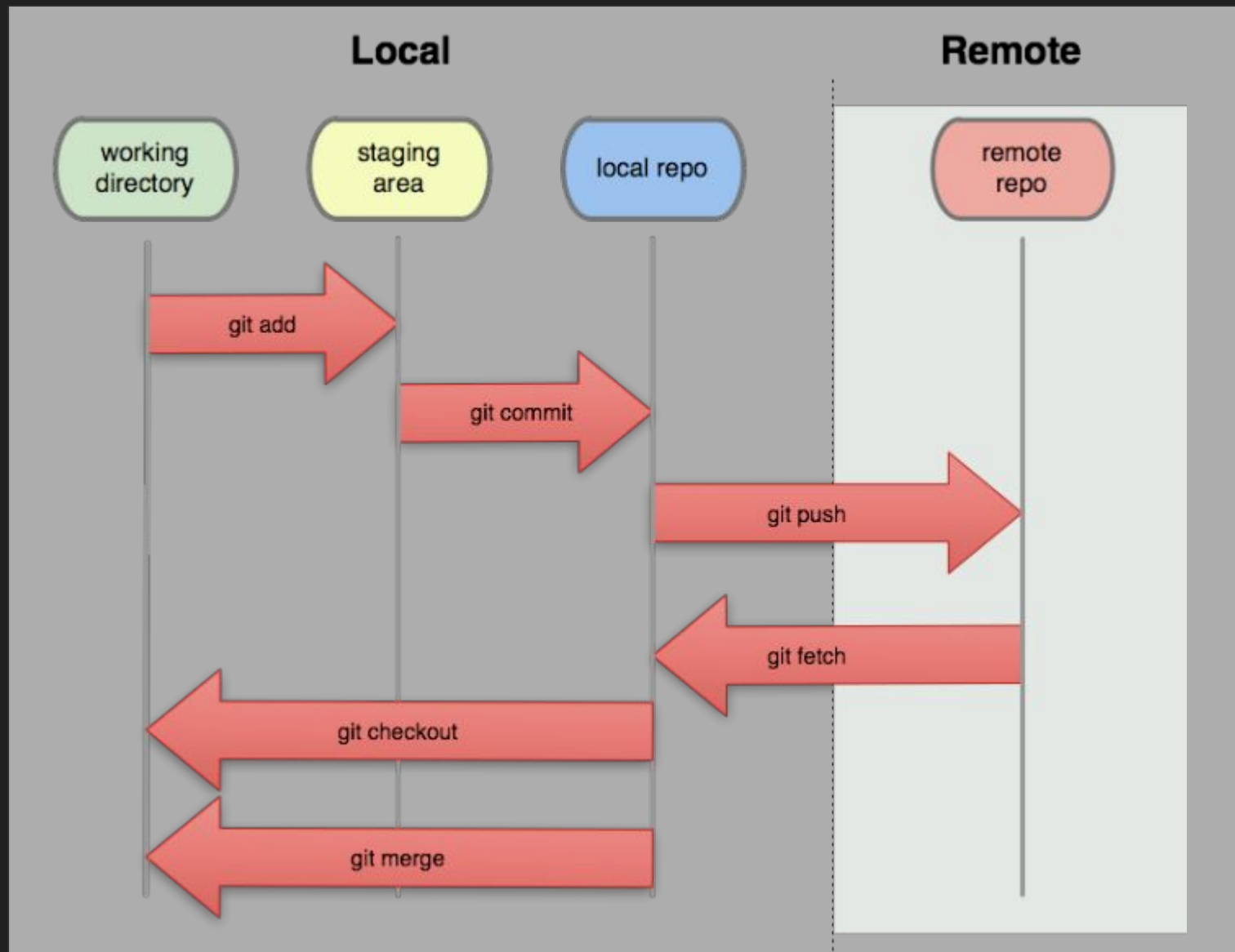
Push when your repo is ready for other people to use/see  
Type `git push -u origin master`

# Songwriting (Demo)

---

Now you can see your updated repo on your Github as well as all the commit messages. You can track your changes every time you push. Check it out on your Github!

# Visual Workflow of Git/Github



# Helpful Links

---

[Setting up Git user.name](#)

[Setting up Git user.email](#)

[Understanding Git: Repos](#)

[Learn the Command Line \(codecademy\)](#)

And of course, Google is your best friend! Most of this stuff is online <3

# Thank you!



[opensource@oxy.edu](mailto:opensource@oxy.edu)



[oxyopensource](https://www.instagram.com/oxyopensource)



[oxyopensourceclub](https://www.facebook.com/oxyopensourceclub)