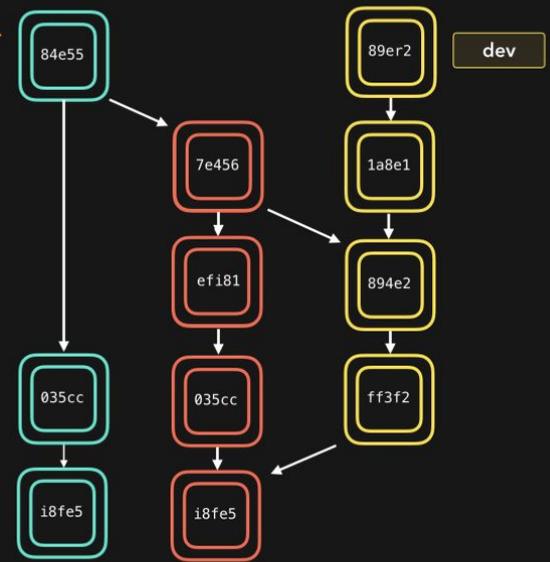


```
children: [  
  icon(icon, color: color  
  container(  
    margin: const. EdgeIns  
  child:  
    label  
    style
```

 Google Developer Student Clubs
University of Toronto Mississauga

Git Workshop + Open Source Initiative 2023



Abdullah, Ido

Introduction to Git

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

What is it?



Git is a distributed **version control system** that enables teams of software engineers to **collaboratively** work on repositories of code

Version Control System...?



Analogy: Imagine you're writing a story.

Every time you write or change something in your story, you save it as a new draft. After a few days, you might have several drafts saved. Each draft represents a specific point of the story.

Key idea: each “draft” is a “version” of your story.

Version control system: Git as a version control system keeps track of all these “drafts” of your story – allowing you to go back to them at anytime.

Collaboration...?

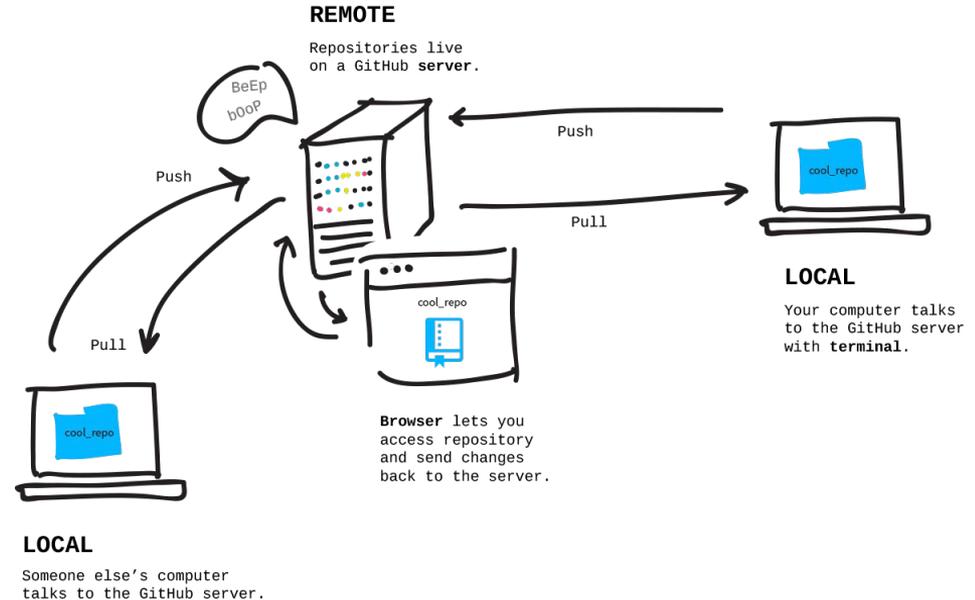


Analogy: Now, imagine you're co-writing this story with friends.

Each of you is making changes and additions. How do you ensure that everyone's changes are captured and nobody overwrites someone else's work? – Git handles this for you!

How collaboration works

- You typically have a few local repositories
- And a centralized remote repository, i.e Github
- A local repository can “push” code into remote
- The rest of the local repositories can pull these changes from the remote.



Credits: https://jlord.us/git-it/challenges/remote_control.html

To summarize...



Git's features include:

- **Backup and restore** of code
- **Synchronization of code** – so multiple people can make changes
- Tracking of changes
- **Undo operations** – so that we can go back to a previous version

And many more!

Okay but how do I use git?

We will be using git using the command line interface in this workshop but there are many GUI versions as well.

Though, we'd recommend using CLI until you are familiar with how git works

Common commands

git clone – copy a “remote” git repository to a new “local” git repository

git init – Create a new git repository from scratch

git add – Allows you to select which files you want to include in your next saved version

More formally, it lets you “stage” files

git commit – Version the files that were added through the “git add” command

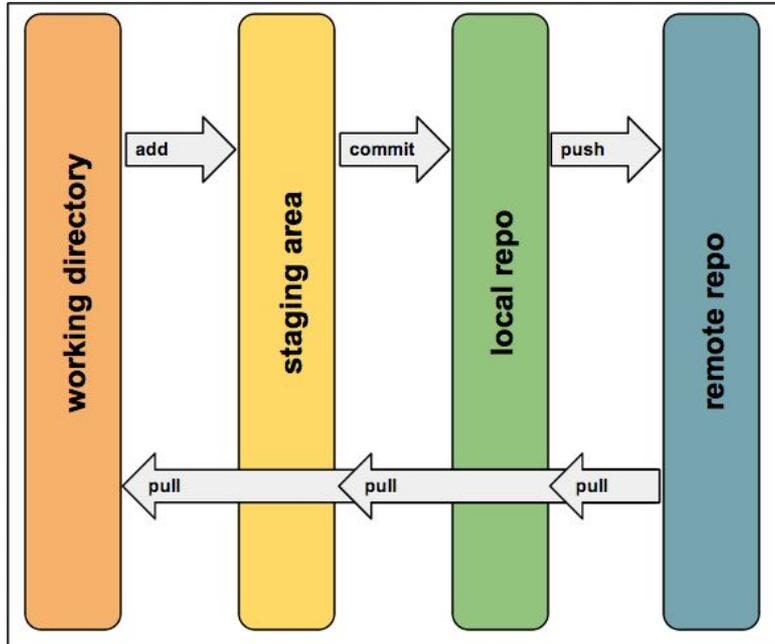
Another way to think of it is that it creates a new version of your code

Common commands

git push – Pushes the state of the “local” git repo to the “remote” git repo. New changes that have been committed locally will be available in the remote.

git pull – Pulls the state of the “remote” git repo to the “local” git repo. New changes that have been committed to the remote will be available locally.

Common commands



Source: <https://www.cureffi.org/2014/08/27/git-tutorial/>

Working with teams can get complicated

When you work in groups you will all be working on the same repository. If you and someone else both want to write to the same files, things can get confusing.

Branching the repository can help.

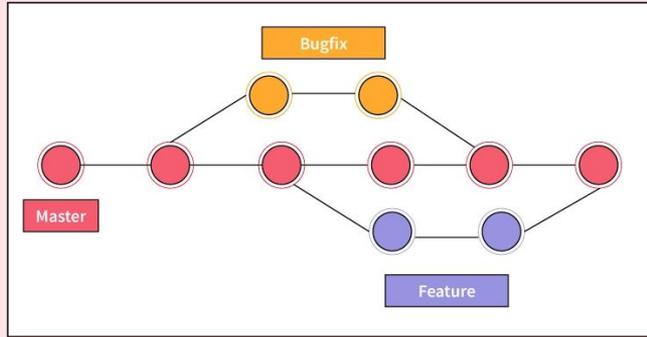
When you branch a repository, you are creating a **isolated sandbox** of it within the repository.

This makes collaboration a lot more seamless.

Branching



Git Branching Strategy



Branching

git branch, to view the current branches in a repository and to see what branch you are on.

git checkout -b <branch name> to create a new branch with a given branch name

git checkout <branch name>, without '-b' flag to switch to existing branches

Remote Server: Github



What is it? A leading web-based platform used for version control and collaboration, allowing users to manage, share, and track changes in their software projects.

Built around Git. Works in conjunction with local Git environments, enabling users to push and pull code between their local repositories into GitHub.

Why do we use it? Widely used for its collaboration based features including tracking changes, reviewing code, project management, and so on!

Open Source

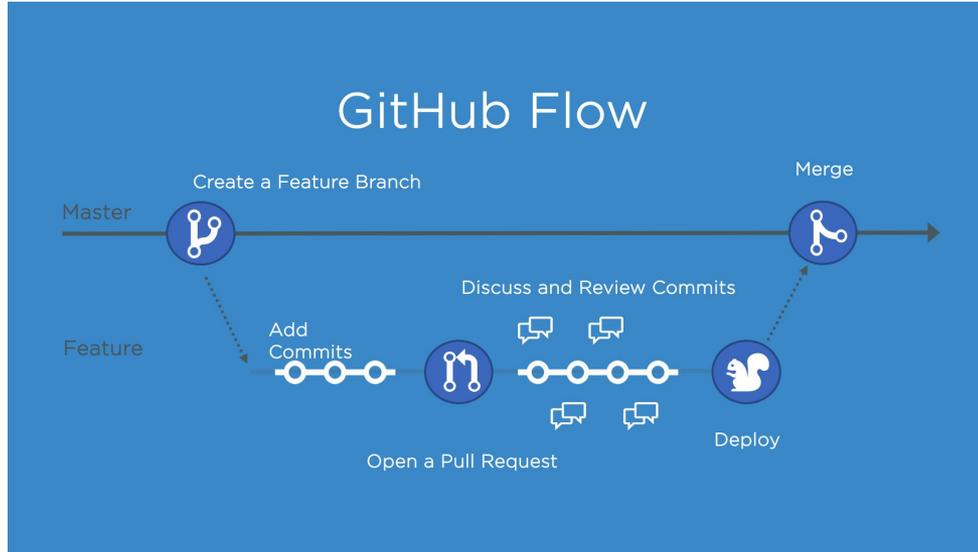
What is it? Open source software is software with source code that anyone can inspect, and modify. It's designed to be publicly accessible and collaborative.

Why open source? Open source projects promote innovation and collaboration. They're often more secure, reliable, and flexible than proprietary software, as they're continuously improved by a diverse community of contributors.

Anyone can contribute to open source projects, regardless of their skill level. Contributions can include writing or improving code, designing graphics, writing documentation, and much more!

GitHub Flow

When contributing to external repositories, you will typically use the following flow:



**We will discuss pull requests and merging in the demo*

Git Workshop Demo

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```




Open Source Initiatives Repositories

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

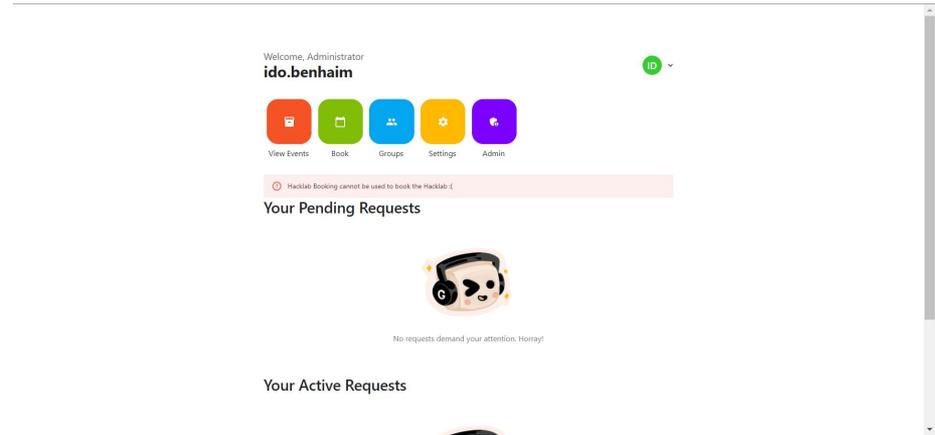
VIAPlanner

VIAplanner is a tool designed by students at the University of Toronto to help the community. We desire to enhance the course selection process.



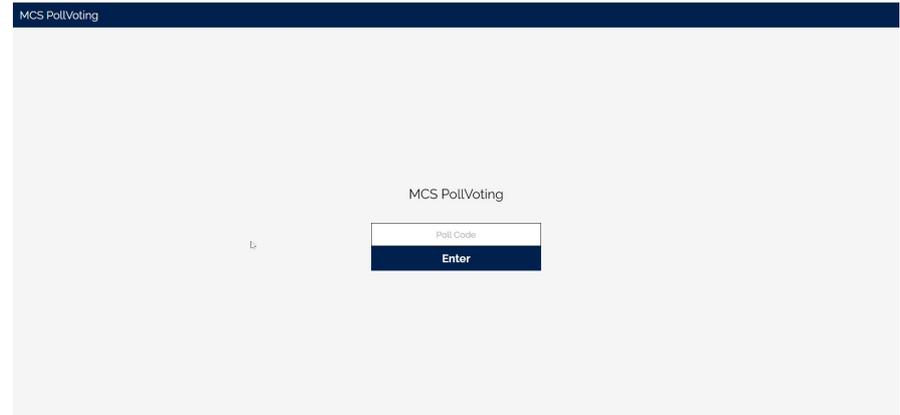
Hacklab Booking

Originally designed to let students book the Hacklab, Hacklab booking is a versatile room booking system.



Poll Voting System

An educational tool designed to be used in classrooms to conduct participation polls.



Course Of Action

Course Manager that uses a script to convert you syllabus pdf to website component with custom notification to keep you on track to get a 4.0



GDSC Website

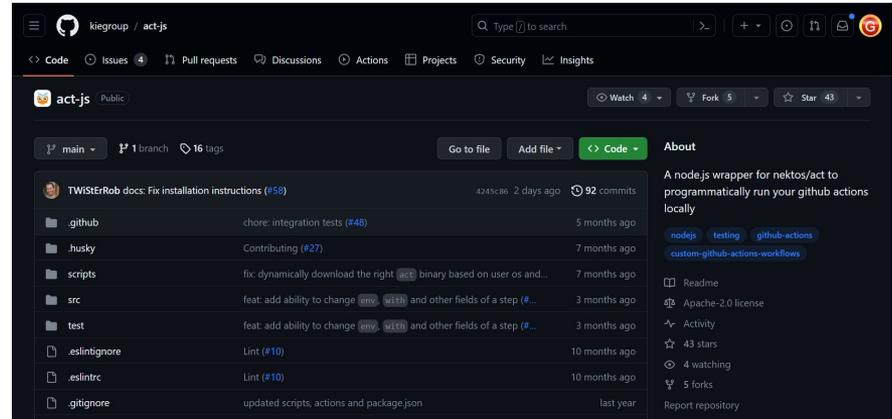
Help improve our club's website :)



act-js and mock-github

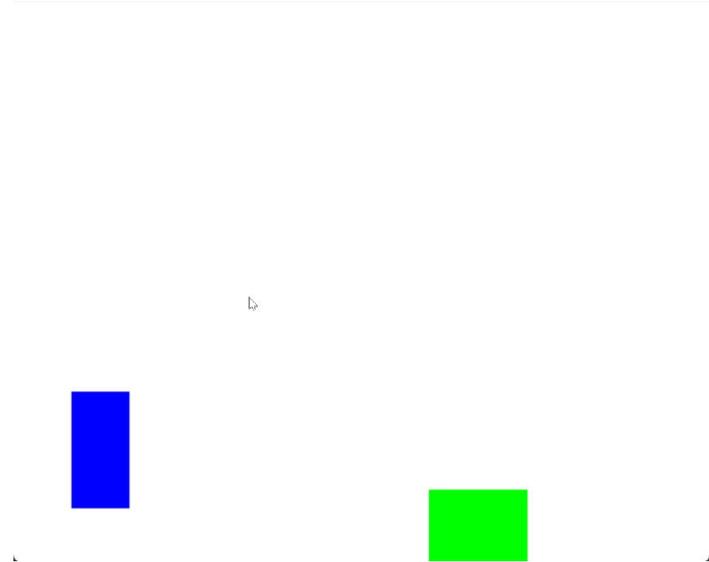
act-js A node.js wrapper for nektos/act to programmatically run your github actions locally.

mock-github - A library to create a local github environment and easily mock github APIs using an octokit like interface



Demo Repository (half points)

A python/pygame recreation of chrome's dino game. This repository is easier to work on and is meant to be beginner friendly.



Points System

Goal: get as many points as possible

Opening a (valid) issue: **10pts**

Fixing an issue: **20pts**

Opening + merging a PR: **5pts**

At the end of OSI (january), your points will be summed up and the winner will win a few cool prizes



Thanks for attending!

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```