# Hacktoberfest Prep: Intro to Open Source and Git

# HELLO!

**I am Sandi Ritter**

I am here because I love to give presentations.

You can find me on slack @sandi or via email sandi@womenwhocode.com

# Open Source Software

What the heck is that???

> *In open source, we feel strongly that to really do something well, you have to get a lot of people involved.*
>
> *Linus Torvalds*

# What is it?

- Open source software is software that anyone can examine, modify, and distribute - the source code is publicly accessible
- It is released under a license that grants users the rights to use, change, and distribute the software to anyone for any purpose

# What are the benefits?

- It is FREE!
- It is flexible - you can alter it to meet your needs
- It is developed in a decentralized and collaborative way by communities rather than a single author or company
- It emphasizes transparency and reliability

# Top Open Source Projects 2019

1. Vue.js
2. React
3. Tensorflow
4. Bootstrap
5. Oh My Zsh
6. Javascript Style Guide
7. D3
8. VSCode
9. Linux
10. React Native

# How is the code managed if anyone can contribute??

# VCS:
# Version Control System

# Version Control Systems

## Local

The simplest form where a database keeps all the changes to files under revision control. Changes are kept as patch sets and you can then add up all the patches needed to re-create a file at any point in time.

## Centralized

A single repository contains all history and each user gets their own working copy of the latest snapshot if the code. You need to commit your changes in the repository and others can only see your changes by updating their working copy.

Example: Subversion

## Distributed

Multiple repositories where each user has their own repository that contains all history in addition to their own working copy. You must commit and push your changes in order to make them visible on the central repository.

Example: Git

git

# Git Setup

1. Download git (if not already done)
2. Set config variables:
   a. git config --global user.name "My Name"
   b. git config --global user.email me@example.com
3. This can be verified by git config --list

# Github

1. Navigate to https://github.com
2. Create an account (if needed)
   a. Complete all fields
   b. Verify email
3. If you have an account, sign in

# Github - Create a Repo

1. Click the "Create Repository" button on the left side of the screen
2. In the Repository Name field, enter *yourusername*.github.io
3. Leave the public setting enabled
4. Click Create Repository button
5. Repo created!

# Github - Clone your Repo

1. Click the copy  button next to the repo name
2. Open your terminal
3. Navigate to a location where you prefer to store your code, or create a new folder
4. Type “git clone thereponameijustcopied” and enter
5. Change directory to the new repo

# Github - Update your Repo

1. Create a basic html file:
   a. echo "Hello World" > index.html
2. Type “git status” - notice the new untracked file
3. Type “git add .“
4. Type “git status” - notice the file is now labeled as a new file

# Github - Commit to Repo

1. Type “git commit -m “initial commit”
2. Type “git status” - nothing to commit, working tree clean
3. Type git push -u origin master
4. Your new file has now been pushed to your repo in github and branch 'master' set up to track remote branch 'master' from 'origin'.
5. Navigate to https://yourusername.github.io.

# Let's Dig in deeper to what we just did

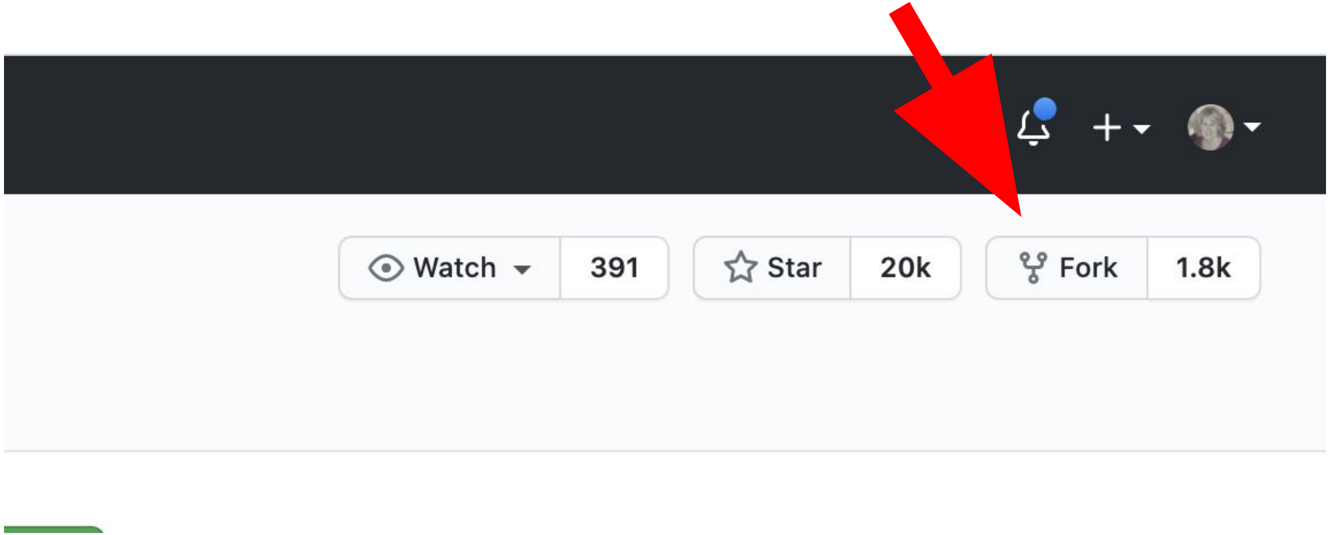
# Git – Clone vs Fork

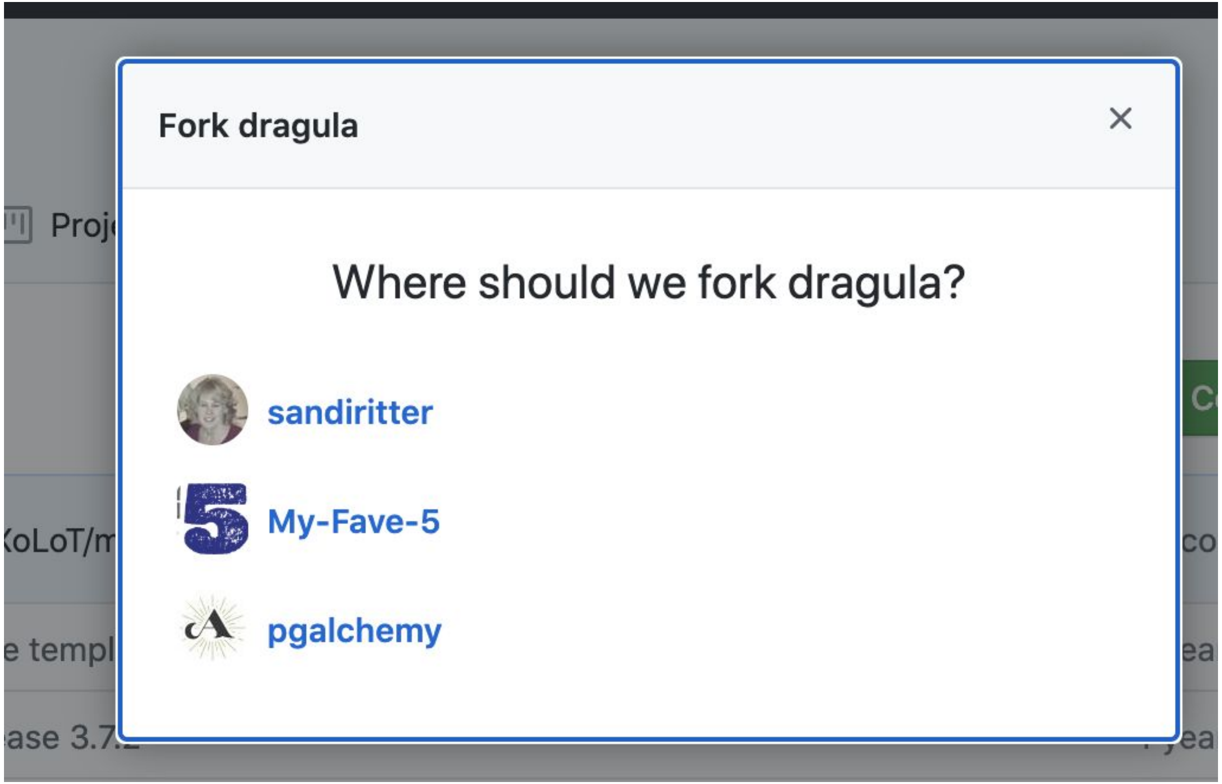
## Clone

A copy of a repo locally on your computer. When you make a clone, you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. The repo you cloned is still connected to the remote version so that you can push your local changes to the remote to keep them synced when you're online.

## Fork

A personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original upstream repository. You can also open a pull request in the upstream repository and keep your fork synced with the latest changes since both repositories are still connected.


Watch
391
Star
20k
Fork
1.8k



Fork dragula ×

Where should we fork dragula?

sandiritter

My-Fave-5

pgalchemy

# Some Useful Terms

## Repo

A project's folder. A repo contains all of the project files and revision history. It can be either public or private

## Checkout

Switches between different versions of the code, and generally is used with branches

## Commit

A revision to the repo. Git creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of the the changes

## Branch

A parallel version of a repository, generally used to change code in an encapsulated environment

## Status

Shows the state of your working directory and helps you see all the files which are untracked by Git, staged or unstaged.

## Pull Request

Proposed changes to a repo submitted by a user. It can be accepted or rejected. You are asking to have your changes “pulled into” the repo

# More Useful Terms

## Add

Adds the change(s) in the working directory to the staging area. However the changes are only recorded once they are committed.

## Push

To send your committed changes to a remote repository on github. You “push” them to the repo

## Fetch

Retrieves the latest meta-data info from the original but doesn’t do any file transferring. Basically checking to see if there are any changes available

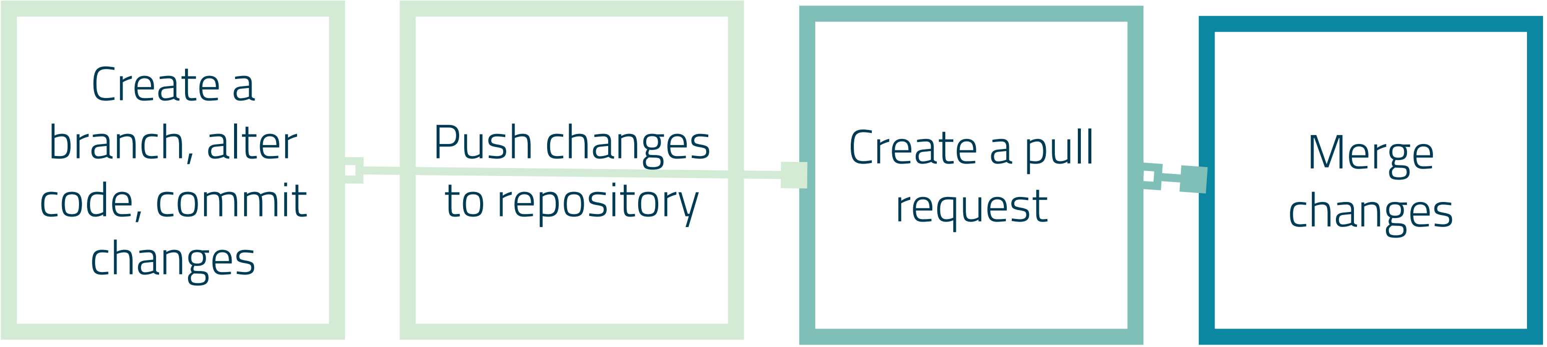
## Merge

Taking the changes from one branch (in the same repository or from a fork), and applying them into another

## Pull

Fetch and download content from a remote repository and update the local repository to match. It’s a combination of two other commands, fetch followed by merge

# Let’s Practice!



Create a branch, alter code, commit changes

Push changes to repository

Create a pull request

Merge changes

# Create a Branch

Navigate to your source code directory

Type ‘git  branch’ to verify you are on the master branch

Type ‘git branch mynewbranchname’

Type ‘git checkout mynewbranchname’

# Make Code Changes and Commit

Open the code in your favorite editor

Make some code changes to your branch - add a photo, change the words, fancy it up!

When done, in your terminal type ‘git status’

Then add the file changes with ‘git add .’

Then verify with ‘git status’

Then commit with ‘git commit -m “my commit message"

# Push Changes and Create a Pull Request

Type ‘git push origin mynewbranchname’

Go to your repo in Github

Click on ‘Pull Requests’

Click on ‘New Pull Request’

Using the compare dropdown, select your branch

Verify the files are as you expect

Click ‘Create Pull Request’

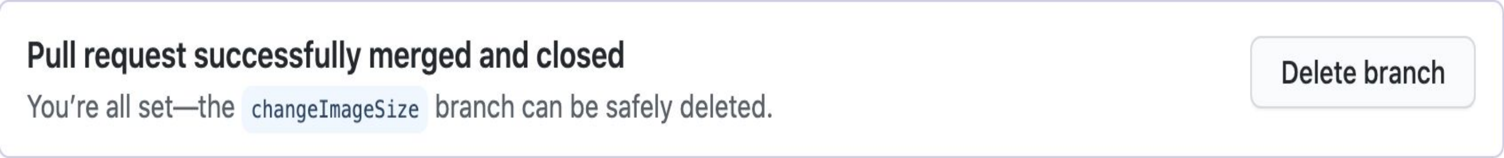
# Merge Pull Request

In your repo in Github,click on 'Pull Requests'

Locate the pull request you just created and select it

Click "merge pull request" button, and "confirm merge"

You should see a message indicating that your changes have been successfully merged



**Pull request successfully merged and closed**

You're all set—the `changeImageSize` branch can be safely deleted.

# What's Next??

# What is Hacktoberfest?

Hacktoberfest is celebration of open source software during October which encourages participation in the open source community

Make four PRs between Oct 1–31 to any public repository on GitHub

Go to Hacktoberfest to register

# How do I find issues for Hacktoberfest

The following resources share repositories that have beginner tasks:

- **Awesome for Beginners**
- **Up For Grabs**
- **Issuehub.io**
- **First Timers Only**
- **First Contributions**

*Check instructions to contribute in each repository*

Source: https://hacktoberfest.digitalocean.com/details

# How do I find issues for Hacktoberfest

You can also search github using the following labels:

- **Hacktoberfest**
- **Good first issue**
- **Beginners**

*Check instructions to contribute in each repository*

# Join the Hacktoberfest FUN!!

- Sign up here
- Join our Cincy WWCode slack channel - #hacktoberfest-2020
- Attend our next meeting on Oct 24th

# Resources

- https://lab.github.com/
- https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf
- https://git-scm.com/docs/git-config
- https://ohshitgit.com/
- https://hacktoberfest.digitalocean.com/details

# THANKS!

## Any questions?

You can find me at:

@sandi (slack)

sandi@womenwhocode.com