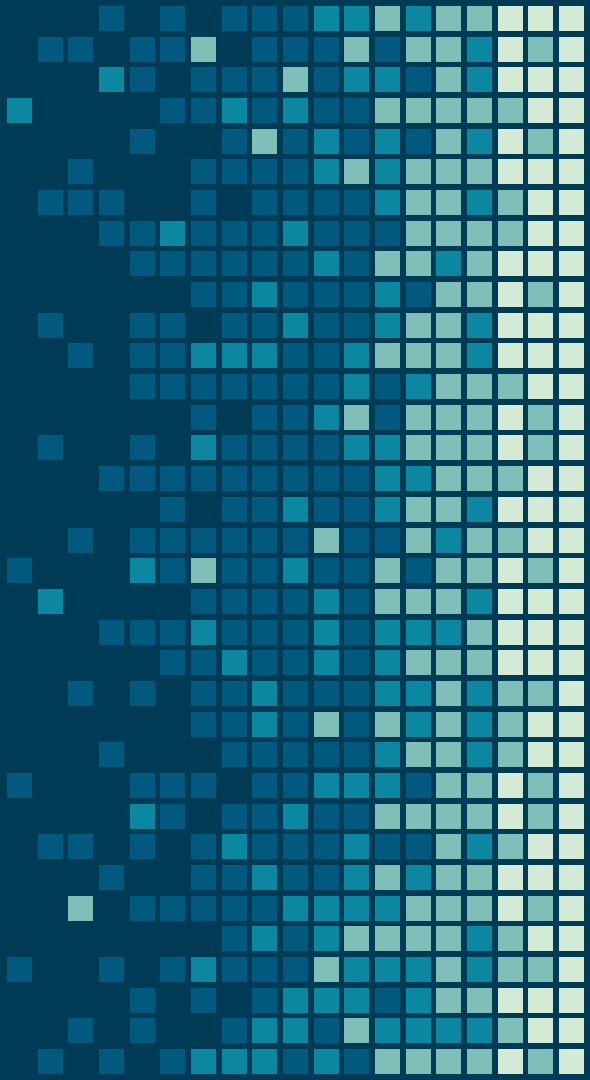
GitHub:
Beyond the Basics



HELLO!

I am Sandi Ritter

- Staff Software Engineer at P&G
- Over 20 yrs experience in IT
- Enjoy exercise and learning to drum

Today's Objectives

- GitHub Specific:
  - Navigating and exploring a GitHub repo
- Git Related:
  - Fork vs Clone
  - Github pages

Today's Goals

- Become familiar with the options available in github repos
- Understanding when to fork and how to create a pull request back into the original repo
- Learn about GitHub pages
- Answer any questions you may have

Housekeeping

- Log in to your GitHub account if you haven't done so already
- Stop me for questions at any time!

GitHub Repository Options

The options available in each repo can vary, but let's start by looking at a few the basics that apply to most repos.

GitHub Issues

An issue in github is a method of tracking work. The work can be a feature request, a task, a bug, an enhancement, etc

Each issue will have a title and a description of what is needed. Each issue can have one person assigns to own it.

Lets review the issues on the VSCode repo

Github Projects

Github Projects is used for Project Management, to aid with organization and prioritization of the work required for a project.

Each Project has a Project Board which is used to track the issues created, where they are in the process flow, as well as notes/comments.

Let’s review the Project Boards found in the bootstrap repo

GitHub Actions

A GitHub action is a small piece of functionality that is used to automate a specific task or workflow. They can perform simple tasks, such as building your code or creating a pull request, or very complex tasks.

They are event driven, meaning they are triggered by some specific activity.

GitHub Actions uses YAML syntax to define the process and the files are always stored in the repository in a directory called .github/workflows

Lets review the actions in the Cincy WWCode repo

GitHub Wiki

Not to be confused with a README file which quickly and briefly tells what your project can do, a GitHub wiki is for hosting documentation about your repo. It can provide additional documentation that goes beyond the scope of a README

Lets review the wiki in the VSCode repo

GitHub Insights

If you enjoy metrics, you will like GitHub insights!

Insights are only available for customers using GitHub Enterprise Cloud. However, it contains so much information that is valuable to be aware it exists, and to poke around in some popular repos.

Lets review the insights in the VSCode repo

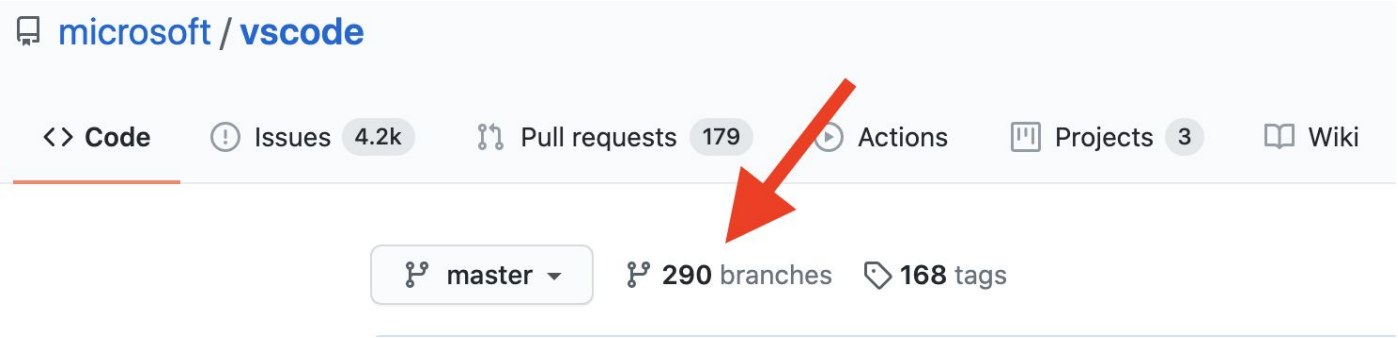
Other Fun GitHub Capabilities

- All things branches
- Tags
- Viewing commits
- Blaming

Let's use VSCode repo to review these items

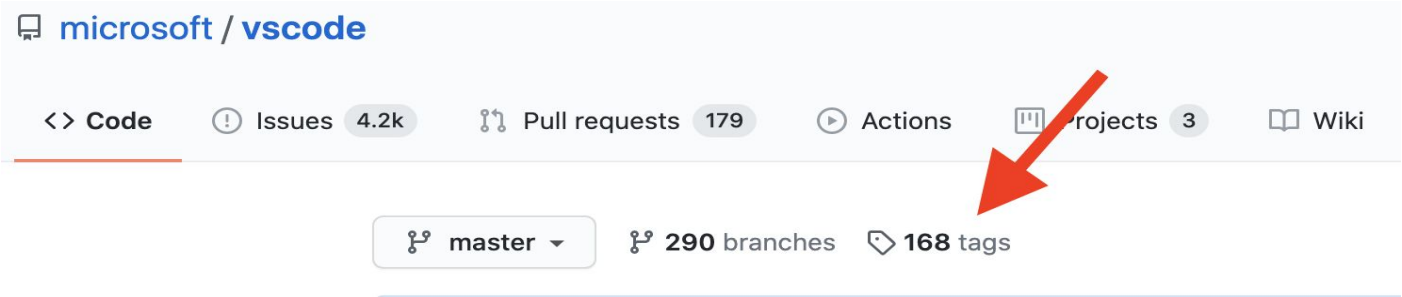
Branches

All branches that have not been deleted can be viewed in the github UI. They can be filtered based on being active, who owns the branch, etc. You can also tell what state each branch is currently in.



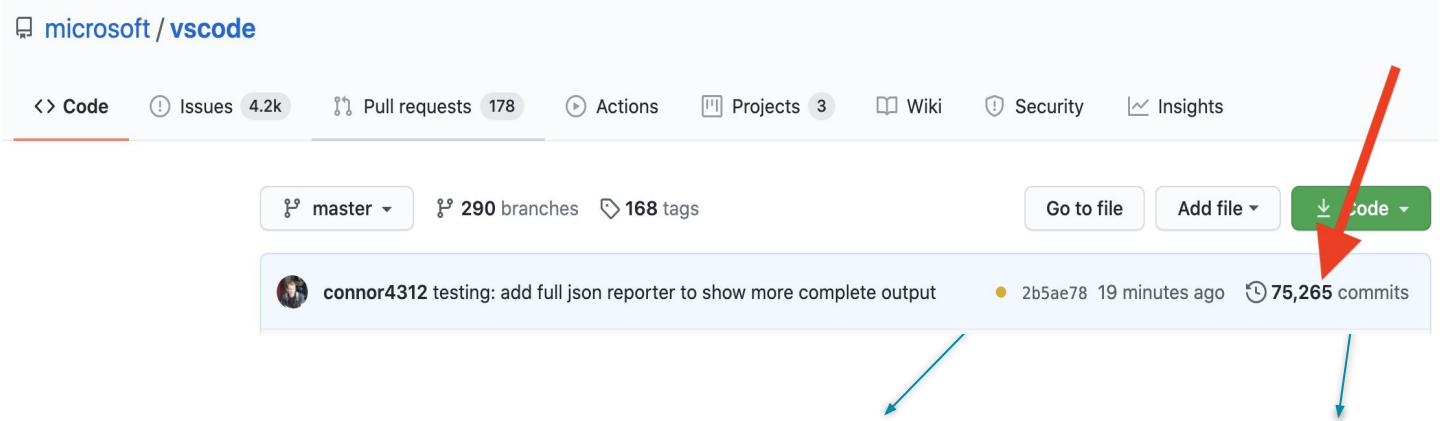
Tags

Tags are used to mark specific points in a repository's history, generally because something important has occurred. The most likely use case for a tag is marking a version release.



Commits

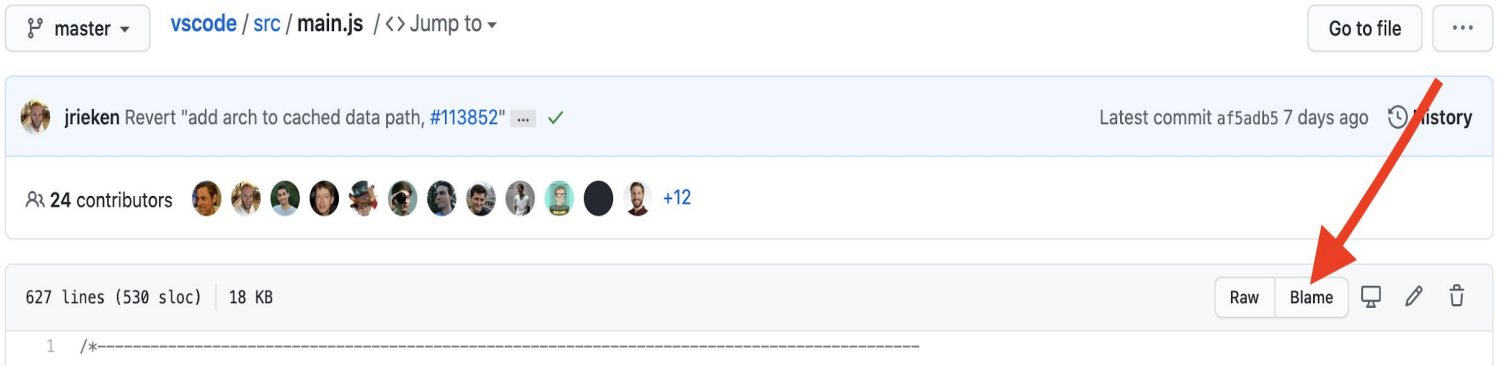
In the GitHub UI, you can view all commits applied to the repo, filtered by branch.



Blame

In the GitHub UI, you can view changes to a file line by line, seeing made the change and also the capability to link to the commit.

To do so, navigate to the main page of the repo, select the file you would like to view and in the upper right corner click Blame



Pulse Check

Git – Fork vs Clone

Knowing when to fork or when to clone is the basis of starting any project using git.

The basic rule of thumb is this - if you DO NOT have write access to a repo, you must start by forking the main repo. Generally speaking, you will only have write access to your own personal repos, and any repos that the owner granted your github user explicit write access.

Git – Fork vs Clone – More Details

Clone

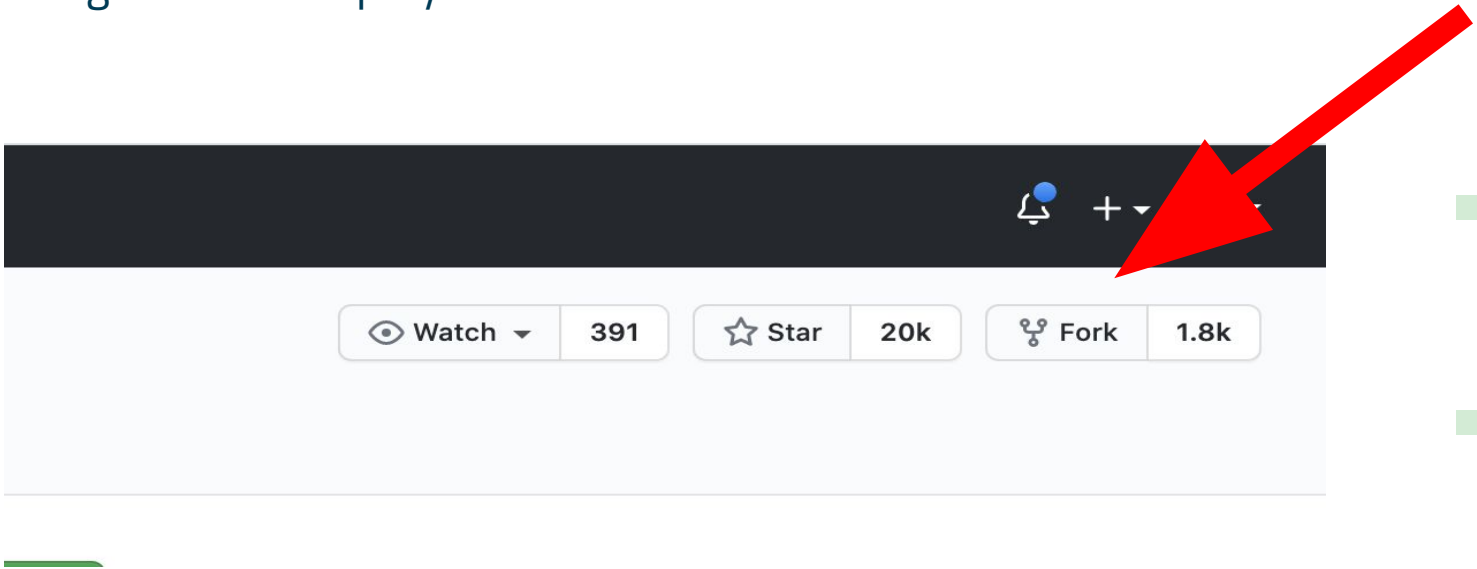
A copy of a repo locally on your computer. When you make a clone, you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. The repo you cloned is still connected to the remote version so that you can push your local changes to the remote to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

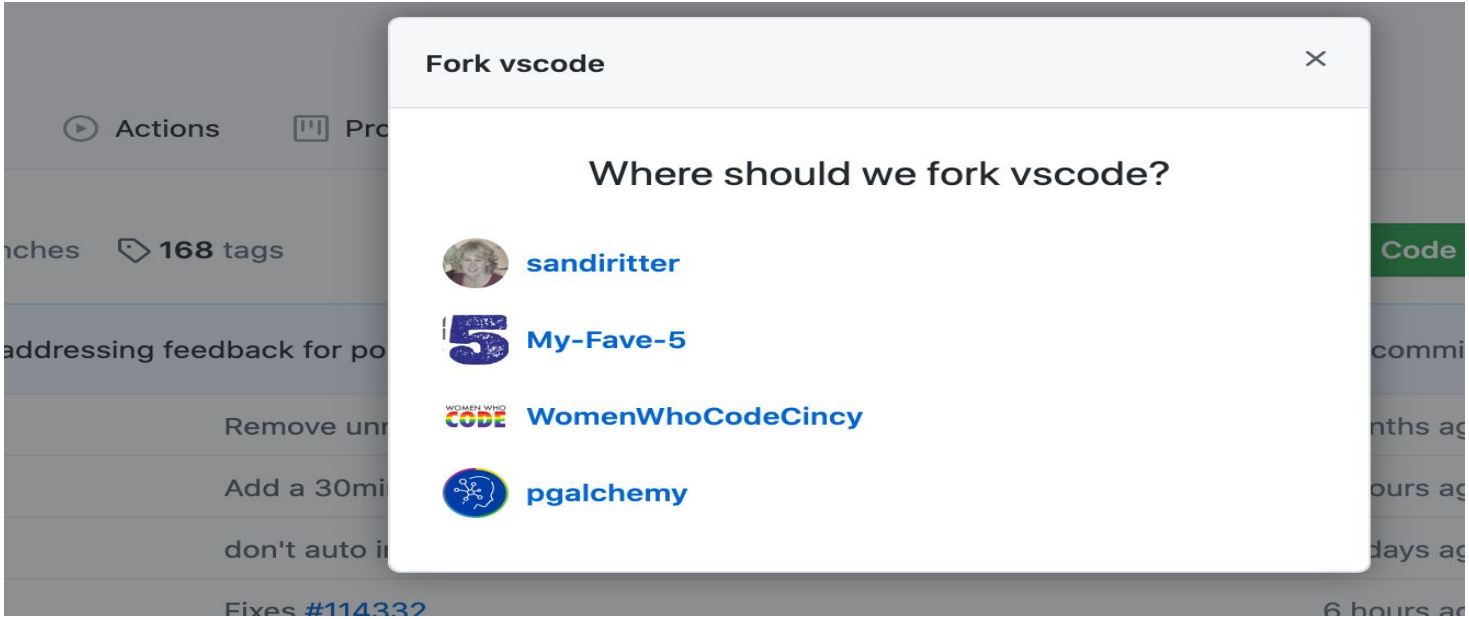
Git - Forking

Navigate to the repo you wish to fork and click the fork button.



Git - Forking

If you have access to multiple repos, you will then have to specify where to fork the repo.
Otherwise it will just fork directly to your repo.



Git – Forking – Making a Change

Now that everyone has their own fork, let's set it up so we can start working.

Start by cloning the repo to your machine:

git clone <therepocloned>

Once cloned, change directory into the repo:

cd <therepocloned>

Git – Forking – Making a Change, cont

Because the repo was forked, we have to ensure that tracking is correctly set up for both repos.

The naming convention used to track your fork is “origin” (this is where your code originated)

The naming convention used for the original repo that was forked is “upstream” (you will send your changes upstream to the main repo)

Git – Forking – Making a Change, cont

Let's start by checking what git automatically sets up for tracking remotes. This command will show you that the origin is already set to your fork, but that the upstream is not set

git remote -v

Now let's set tracking for the upstream repo:

git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPO.git

Now execute *git remote -v* again and both origin and upstream should be set.

Git – Forking – Making a Change, cont

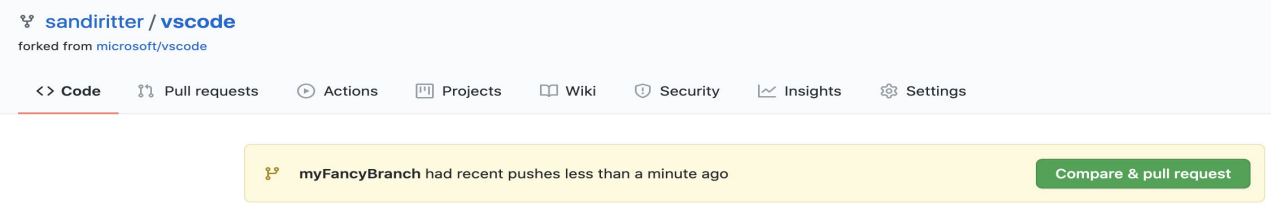
Now that all tracking is properly set, follow the normal flow to create a branch, make a change and push the branch up.

1. Create a branch and check it out:
   a. *git checkout -b myNewBranchName*
2. Using your favorite editor, edit the code as needed
3. Verify your changes and commit them:
   a. *git status*
   b. *git add .*
   c. *git commit -m "a nice concise commit message"*
   d. *git push origin myNewBranchName*

Git – Forking – Making a Change, cont

Let's create a pull request for the branch just created

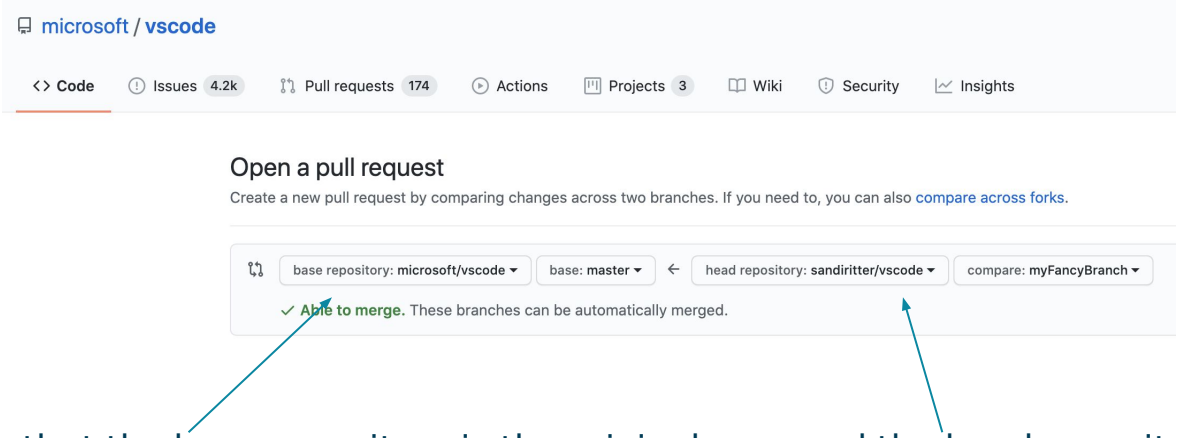
1. Navigate to your repo in github
2. You should see a message box indicating that your branch was recently pushed



3. Click the button to compare and create a pull request

Git – Forking – Making a Change, cont

4. You should now see a screen like this:



Verify that the base repository is the original repo and the head repository is your fork where the branch is the branch you just created.

Also notice in the upper left-hand corner that you are now creating a pull request in the original repo. Click Create Pull Request to complete the process

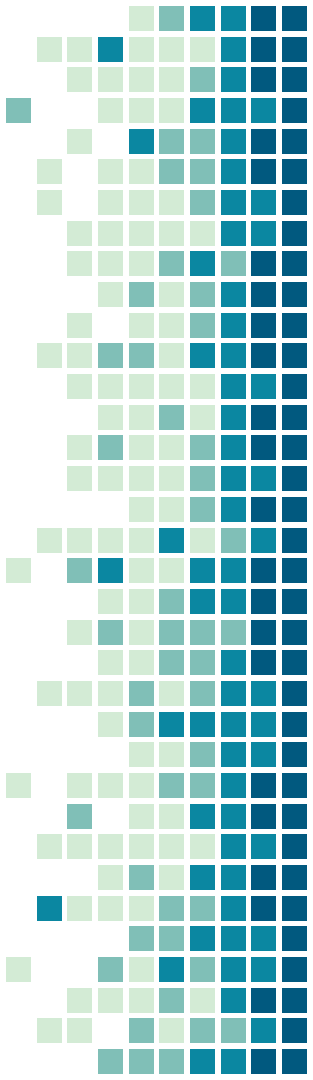
Let's Practice!

Let's take some time to complete Lab One, which will give you the opportunity to fork a repo and create a pull request.

Lab one can be found here - https://sandiritter.github.io/GithubBeyondTheBasics/labs/labOne.html

GitHub Pages

- GitHub Pages are public webpages hosted and published via GitHub. Basically it gives you the opportunity to create your own website
- There are 3 types of GitHub Pages sites: project, user, and organization
- Each GitHub user or organization can have one GitHub page but unlimited sites for projects
- You can create your page either by scratch or you can leverage some templates that github provides
- It’s a fun way to play around and learn new things, or to use as a way to advertise yourself & your skills



Let's Practice!

Let's take some time to complete Lab Two, where everyone will set up their personal GitHub page.

Lab two can be found here - https://sandiritter.github.io/GithubBeyondTheBasics/labs/labTwo.html

Resources

- GitHub Learning Lab
- GitHub glossary
- Git Cheat Sheet
- Forking Projects
- Free Udemy Course on git
- Github Pages

THANKS!

Any questions?

You can find me at:

sandiritter68@gmail.com