

Collaboration: Understanding Git



This is an amazing section title

What is Git & Why Do We Use It?

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency."

-git-scm.com

Git is the tool that allows the software engineering community the ability to collaborate across countries, business sectors, and more specifically, companies.



Getting Started With Git

Getting Up and Running with git is pretty straightforward. Each project (repository) will start with the following basic steps when creating from scratch.

Start a new Repo (Local)

git init

Add New Files To Current Repo (Local)

git add
<name-of-file(s)> OR
period (.)

Commit New Files Into Holding Area

git commit -m "First Commit"

Add A New Remote (Local)

git remote add

<name-of-remote>

<URL-To-Remote-Repo>

Send New Changes To Remote Repo

git push
<name-of-remote>
<name-of-branch>

Here are the steps in sequence

To add a bit more context to the previous slide, here's step-by-step breakdown of working with git

**The Steps listed below assume a project
(repo) has been created on Github.com**





Github Collaboration

Collaborating with teams across the world



Using Github.com - A Collaboration Tool

Social Coding

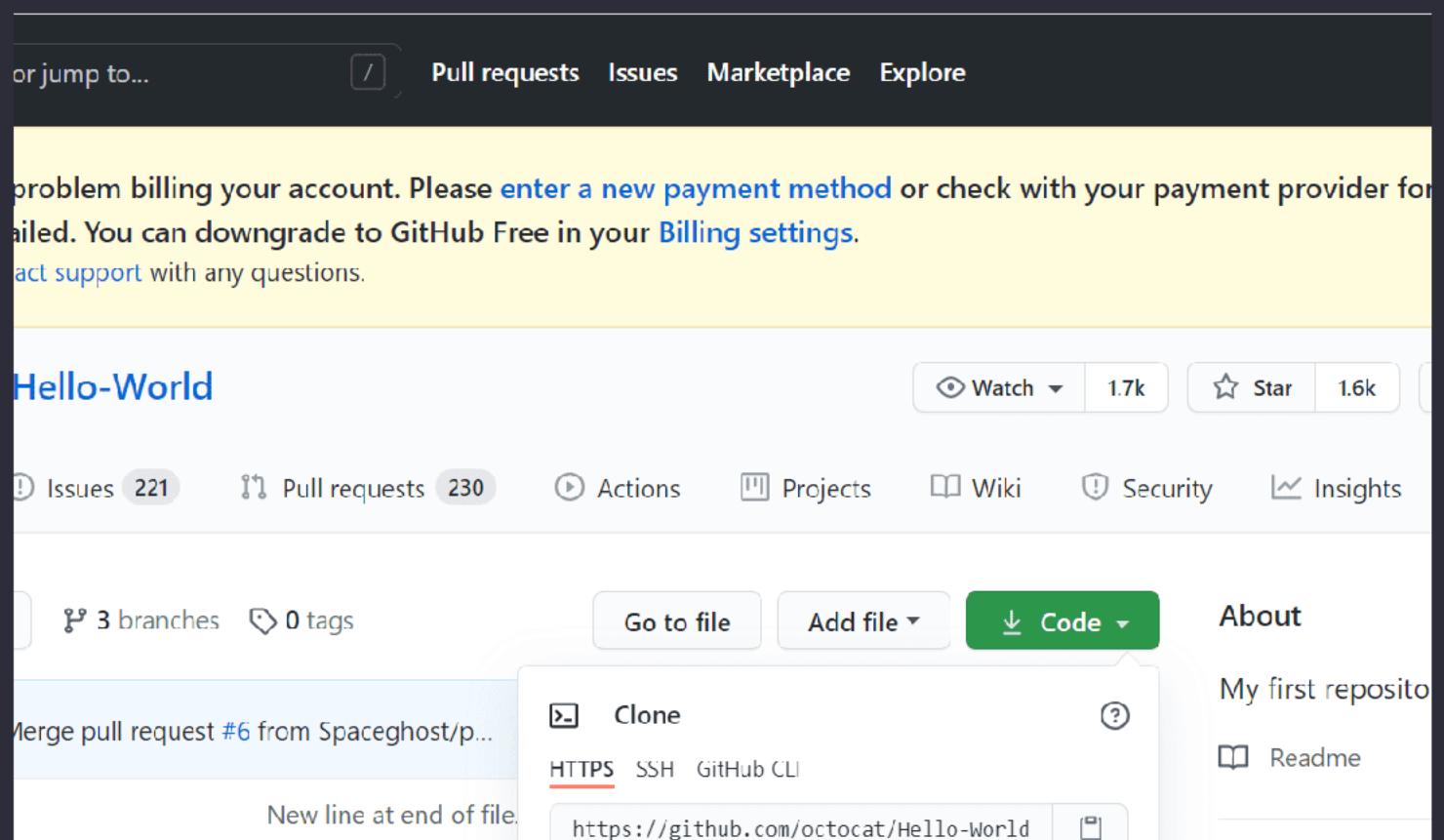
Although Git and Github are said within the same breath most of the time, they are different.

GIT - The Version Control tool

GITHUB - The Platform for sharing code

This is an amazing slide title

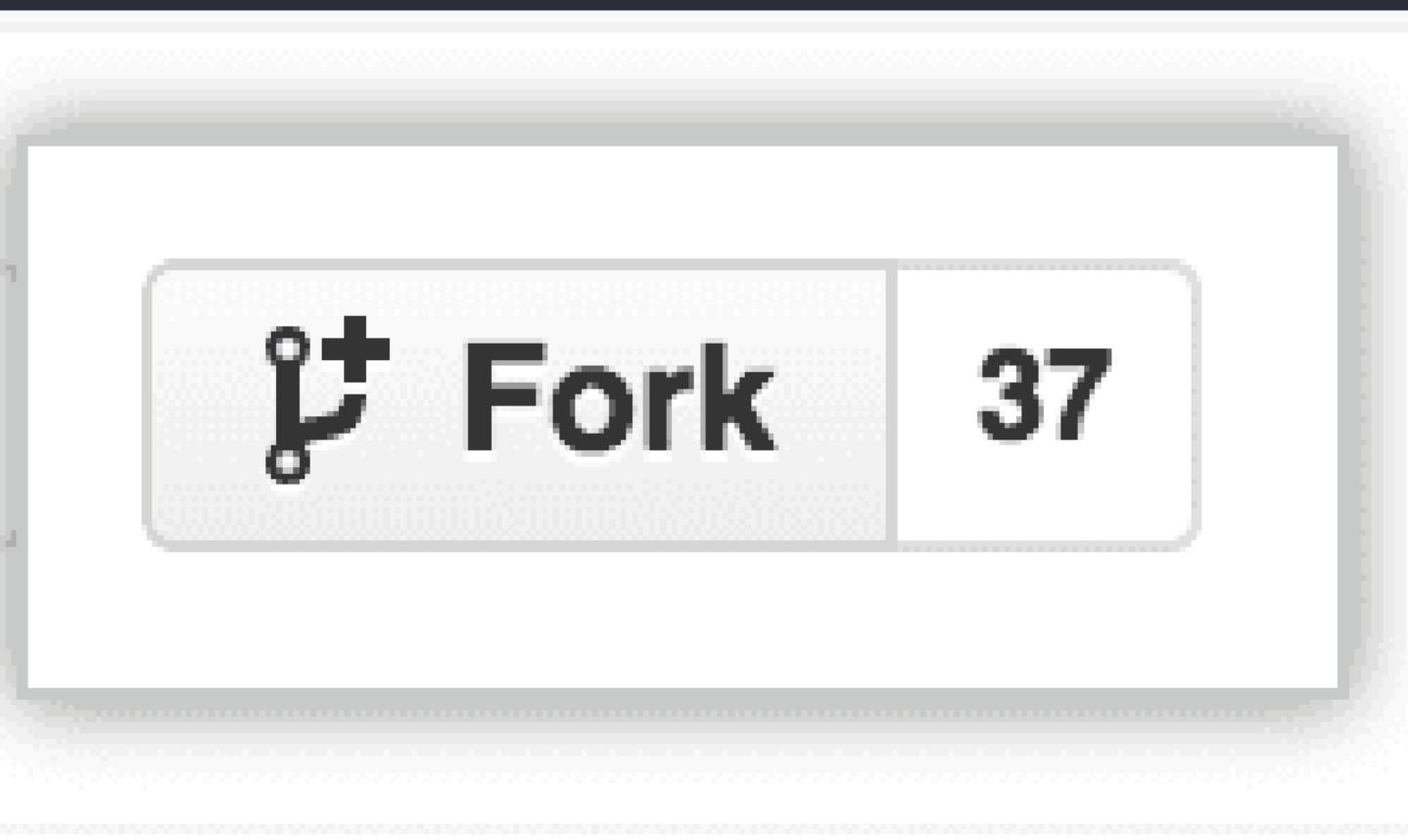
Start By Navigating to a Repo
For Example: <https://github.com/octocat>Hello-World>



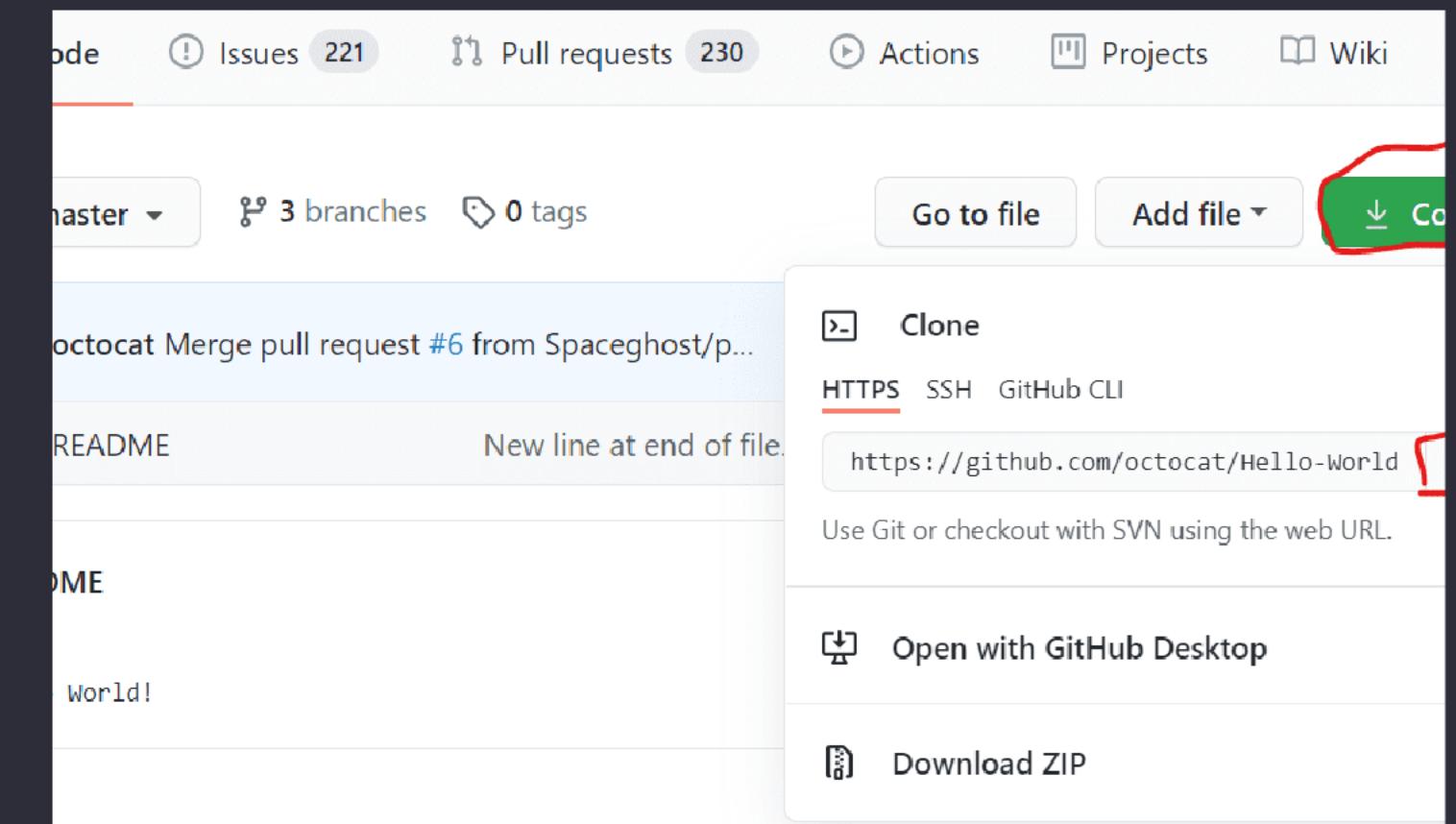
Make A Change To A File

A screenshot of a code editor showing a file named 'queries.py'. The code contains several commits, with one commit highlighted in yellow. The commit message says 'Merge pull request #6 from Spaceghost/p...'. The code editor interface shows the file structure on the left and the code content on the right.

Then Fork the repo



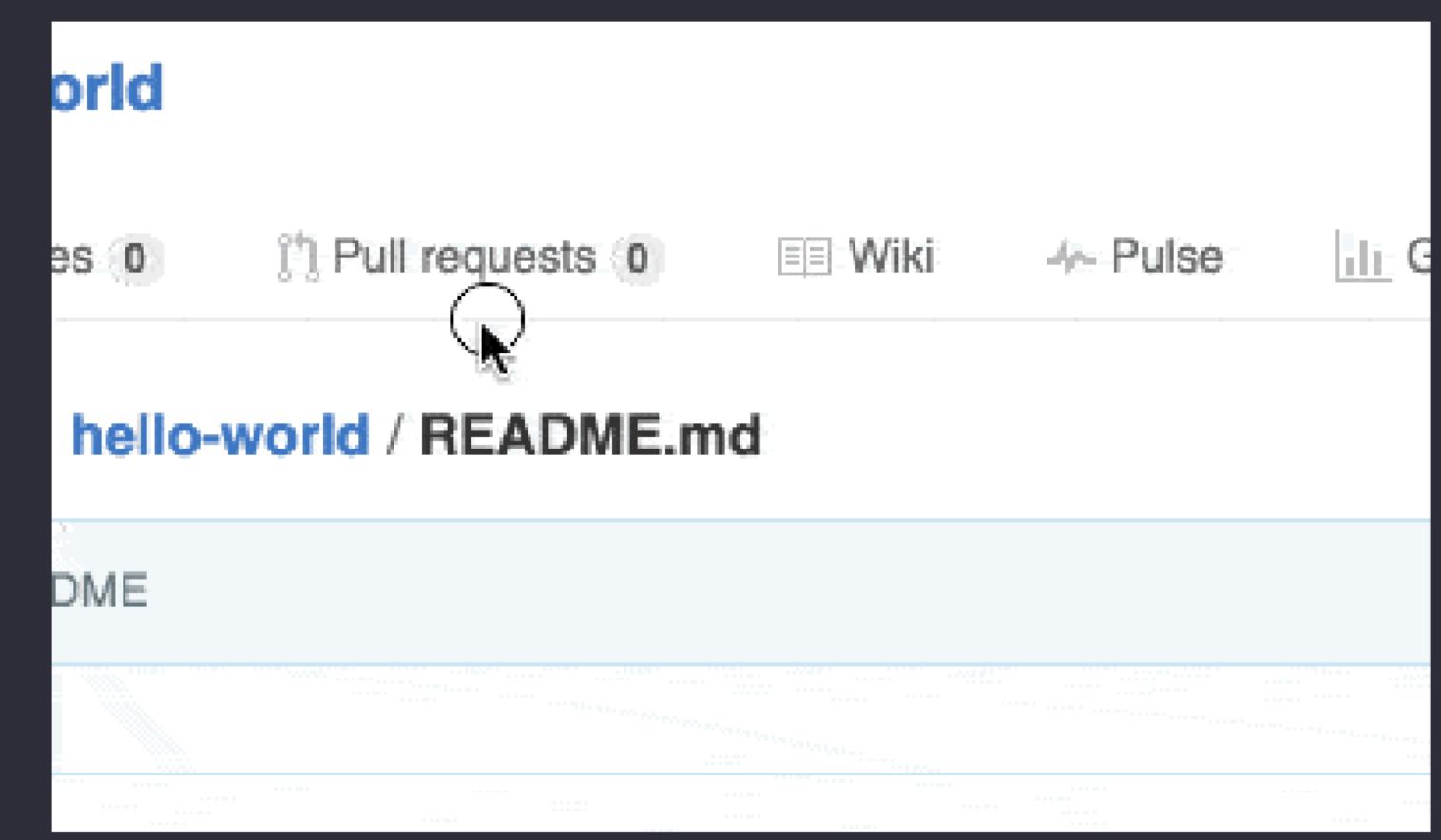
Then Clone the repo



Commit New Changes



Create Pull Request



Creating A New Branch

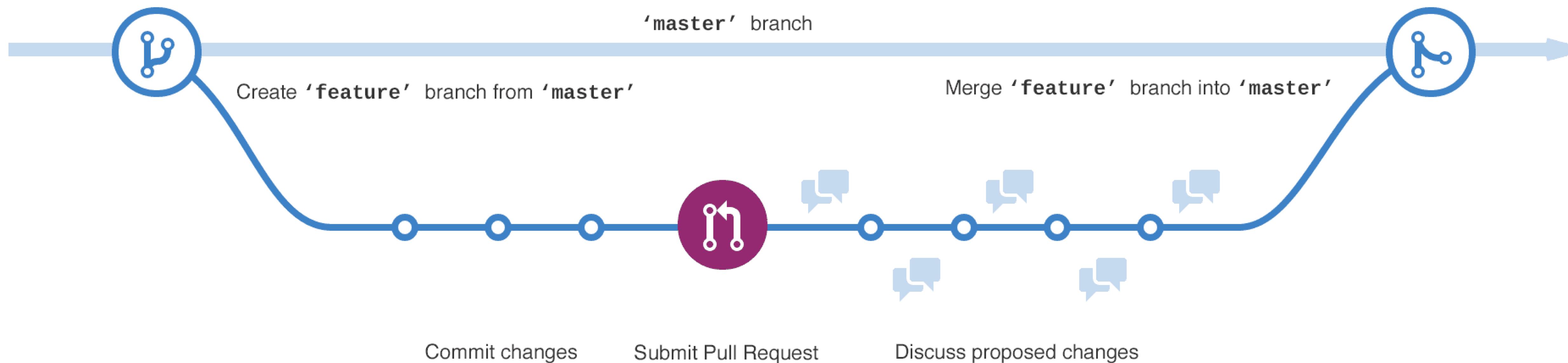
Seperating Code By Branches

Using Git Branches

A Repo branch is a way to separate code without overriding previous code.

This allows for testing of new features without introducing regression

To create a git branch (Local) -- **git checkout -b <Name-Of-Branch>**
[This command auto switches into the new branch]



This is an amazing section title

Let's Put This To Work!

With a high level overview of git in your back pocket, let's put this into practice.

BTW, there are 2 ways
to get this done!

```
    .keyframes ripple {
        from {
            width: .1%;
            height: .1%;
            opacity: 1;
        }
        to {
            width: 100%;
            height: 100%;
            opacity: 0;
        }
    }
```

Helpful Git Resources

The following displays helpful links while working with git & github

Git hello World Guid

<https://guides.github.com/activities/hello-world/>

Getting Started with Github Pages

<https://guides.github.com/features/pages/>

Github Cheatsheet

<https://education.github.com/git-cheat-sheet-education.pdf>

Getting Stared With Forking

<https://guides.github.com/activities/forking/>

Getting Started with VS Code &

Github

<https://code.visualstudio.com/docs/editor/github>

ALL GITHUB GUIDES

<https://guides.github.com/>