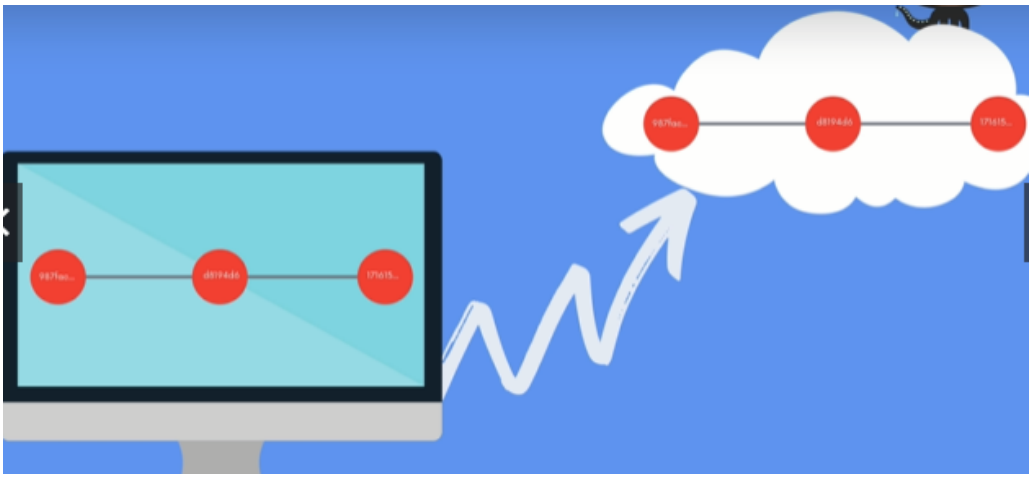
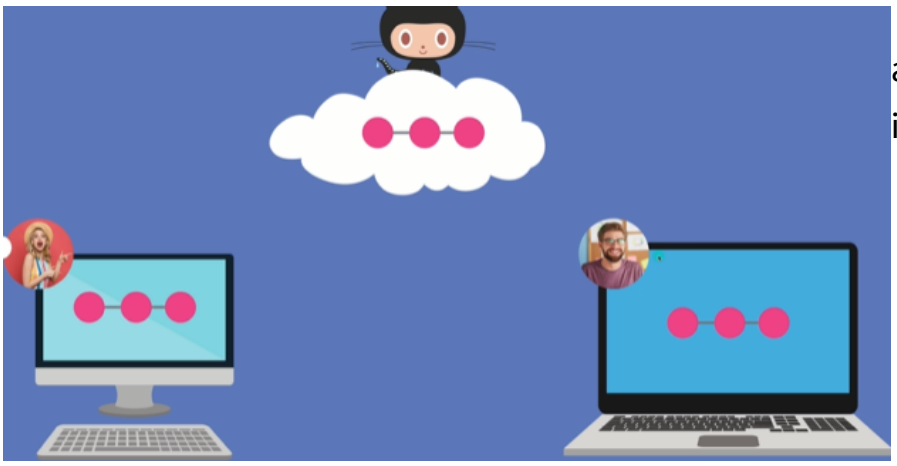
GITHUB

Its difficult to answer that question, cuz over the year it has add bunch of features.



1.backup.

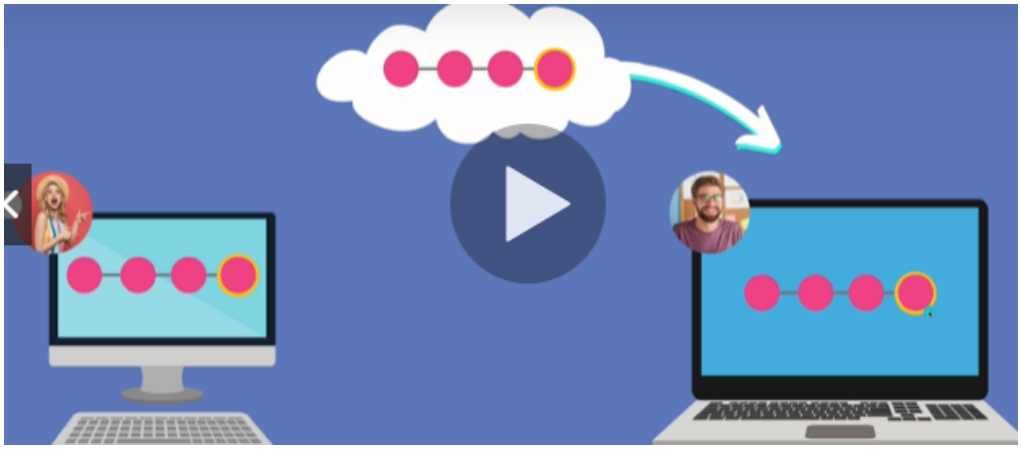
2. collaborate

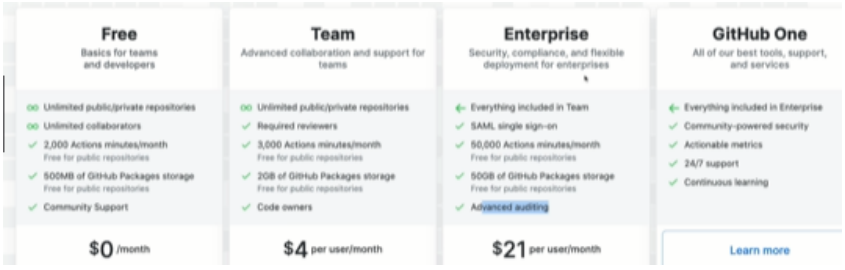


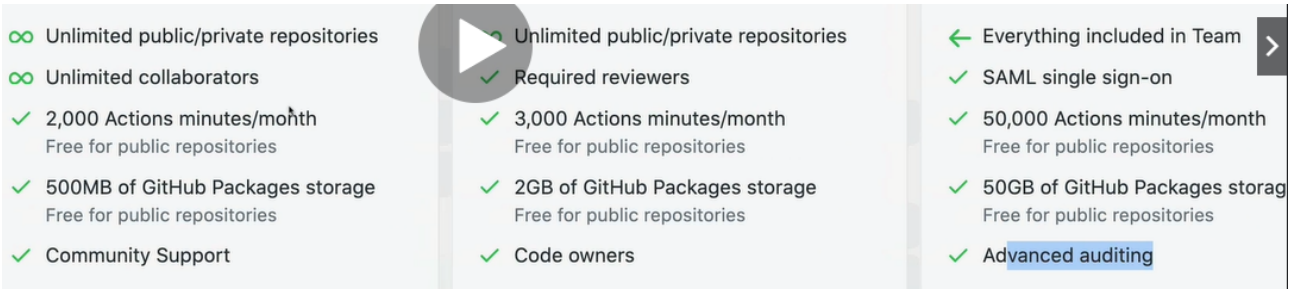
in left diagram, zelda is an engineer, and lincoln is graphic designer.

Using github they can all clone and get access to this repositories in their machine and they can make contributions and put it on github and pull their contributions.

Or commits.







Social network for git repos, you can chat, write comments, approve , disapprove, suggest changes to projects , merge code in , reject code , call attention to certain lines of code.

Interactive developer enviorment : codespaces. Additional features like automatically scanning your repositories and looking for common vulnerbalities.

And letting you know ahead of time.

Why You should Use Github

For collaboration, backup, resume, trying to meet potential mentors.

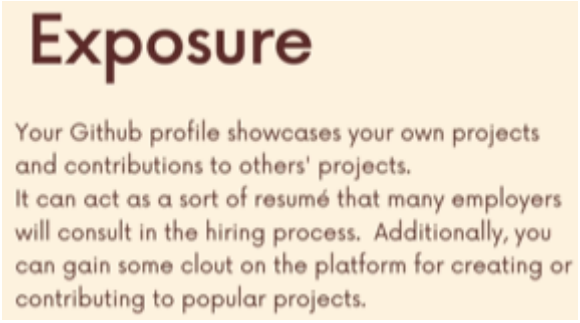
Open Source Projects:

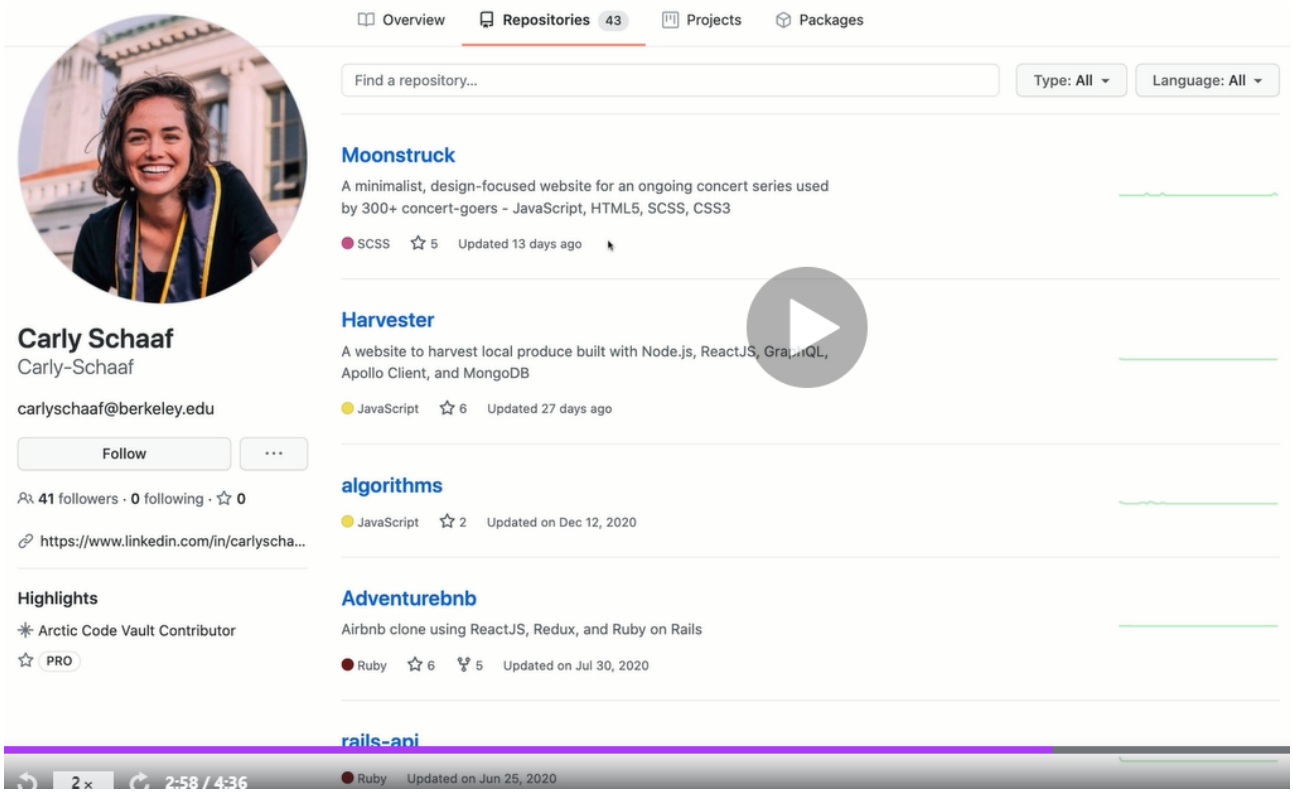
is where the codebase is available. Not just to typically (for viewing and using) but also contributions. To Fix bugs and to Discuss with greater community of ppl.

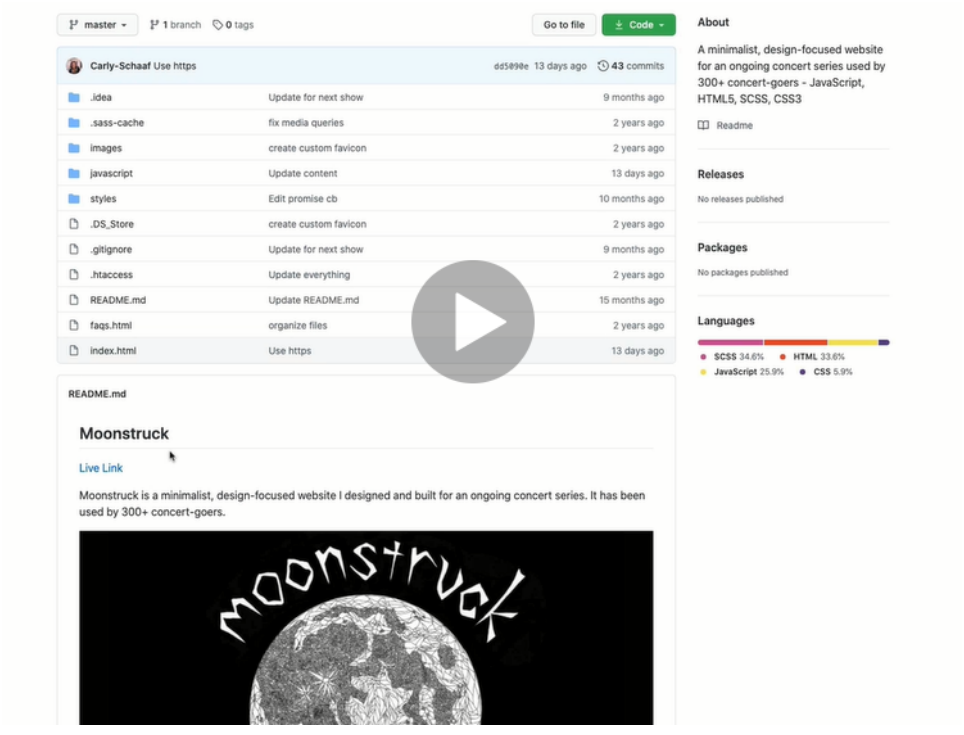
Basically the users of the product. Github is the home of Open source projects over the internet today.

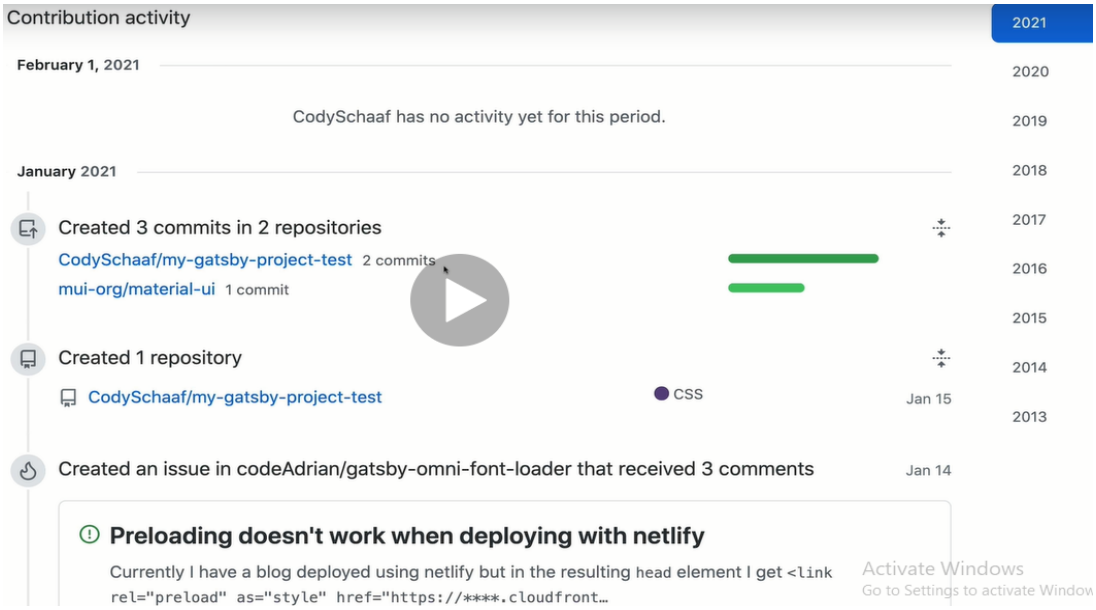
Open source contribution is a great prove that you know what you doing.

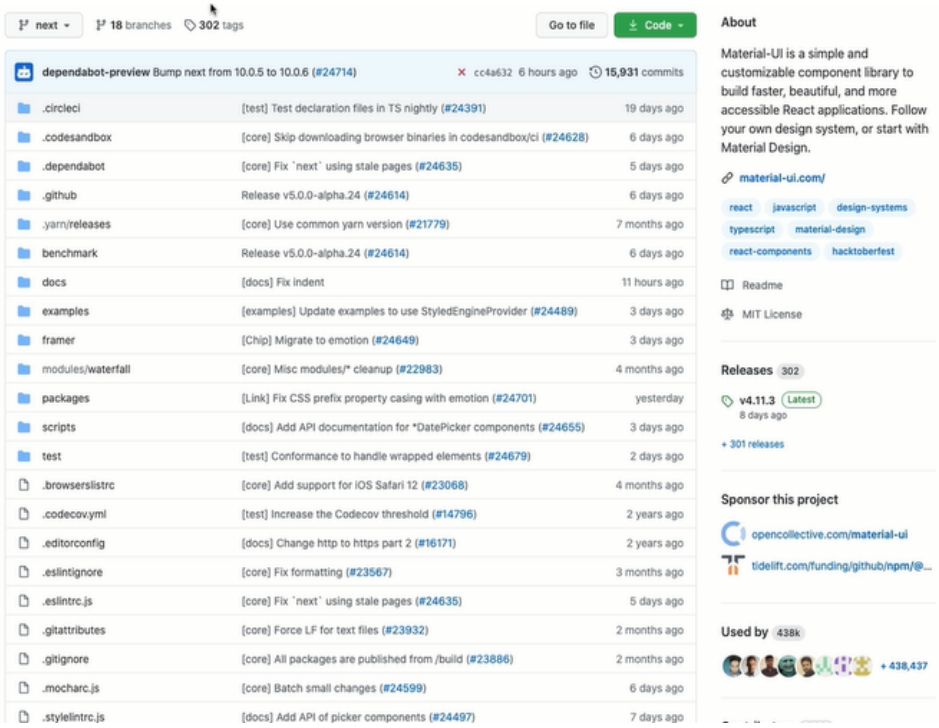
For e.g: React, Tensorflow.




Overview
Repositories 43
Projects
Packages
Find a repository...
Type: All
Language: All
Carly Schaaf
Carly-Schaaf
carlyschaaf@berkeley.edu
Follow
41 followers · 0 following · 0
https://www.linkedin.com/in/carlyscha...
Highlights
Arctic Code Vault Contributor
PRO
Moonstruck
A minimalist, design-focused website for an ongoing concert series used by 300+ concert-goers - JavaScript, HTML5, SCSS, CSS3
SCSS 5 Updated 13 days ago
Harvester
A website to harvest local produce built with Node.js, ReactJS, GraphQL, Apollo Client, and MongoDB
JavaScript 6 Updated 27 days ago
algorithms
JavaScript 2 Updated on Dec 12, 2020
Adventurebnb
Airbnb clone using ReactJS, Redux, and Ruby on Rails
Ruby 6 5 Updated on Jul 30, 2020
rails-api
Ruby Updated on Jun 25, 2020
2 ×
2:58 / 4:36


master
1 branch
0 tags
Go to file
Code
Carly-Schaaf Use https
dd5090e 13 days ago
43 commits
.idea Update for next show 9 months ago
.sass-cache fix media queries 2 years ago
images create custom favicon 2 years ago
javascript Update content 13 days ago
styles Edit promise cb 10 months ago
.DS_Store create custom favicon 2 years ago
.gitignore Update for next show 9 months ago
.htaccess Update everything 2 years ago
README.md Update README.md 15 months ago
faqs.html organize files 2 years ago
index.html Use https 13 days ago
About
A minimalist, design-focused website for an ongoing concert series used by 300+ concert-goers - JavaScript, HTML5, SCSS, CSS3
Readme
Releases
No releases published
Packages
No packages published
Languages
SCSS 34.6%
HTML 33.6%
JavaScript 25.9%
CSS 5.9%
README.md
Moonstruck
Live Link
Moonstruck is a minimalist, design-focused website I designed and built for an ongoing concert series. It has been used by 300+ concert-goers.
moonstruck





Stay Up To Date

Being active on Github is the best way to stay up to date with the projects and tools you rely on. Learn about upcoming changes and the decisions/debate behind them.

Cloning Github Repos With Git Clone

Cloning

So far we've created our own Git repositories from scratch, but often we want to get a **local copy of an existing repository** instead.

To do this, we can clone a remote repository hosted on Github or similar websites. All we need is a URL that we can tell Git to clone for use.

git clone

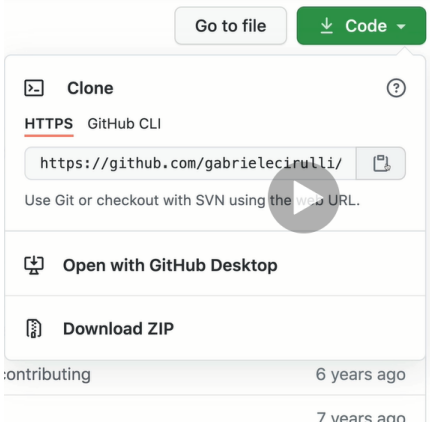
To clone a repo, simply run **git clone <url>**.

Git will retrieve all the files associated with the repository and will copy them to your local machine.

In addition, Git initializes a new repository on your machine, giving you access to the full Git history of the cloned project.

```
❯ git clone <url>
```

Make sure you are not inside of a repo when you clone!




```
Cloning ❯ ls
Cloning ❯ git clone https://github.com/gabrielecirulli/2048.git
Cloning into '2048'...
remote: Enumerating objects: 1315, done.
```

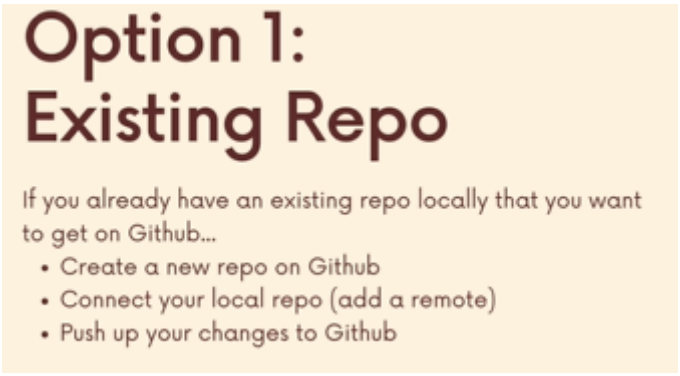
```
README.md       index.html
2048 ❯ git status
On branch master
Your branch is up to date wit

nothing to commit, working tr
2048 ❯ git log
```

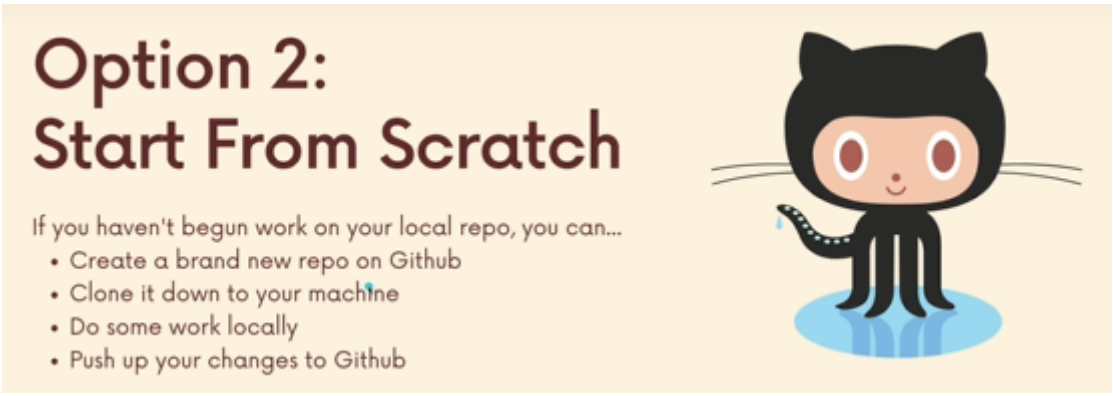
Cloning Non-Github Repos

git clone works with any type of repositories.

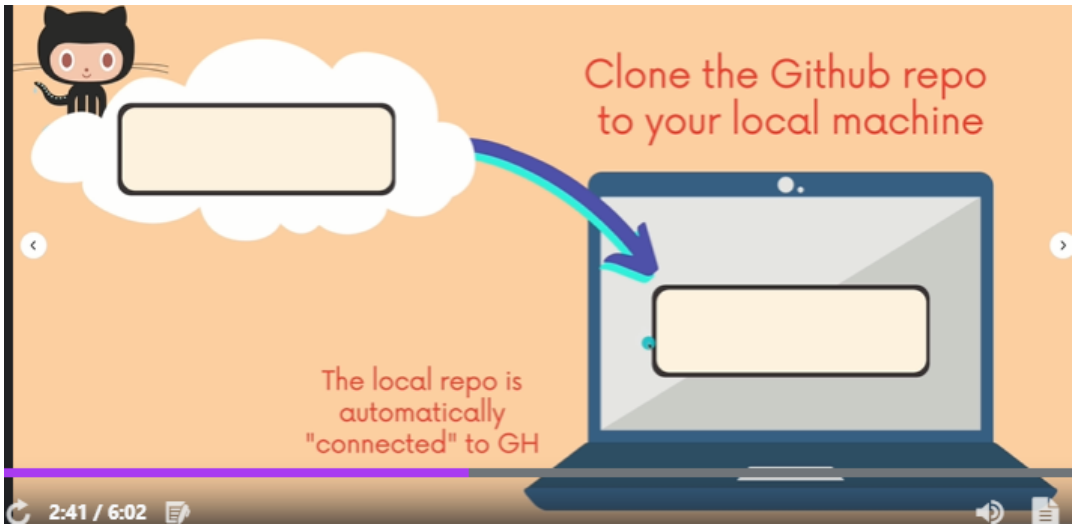
Github Setup: SSH Config

Creating Our First Github Repo!





If i do that way then i dont have to manually connect my local repository to my github repository. If i clone this repo from github its automatically connected to that github url.



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/IDK9911/to_learn_git.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# to_learn_git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/IDK9911/to_learn_git.git
git push -u origin main
```

...or push an existing repository from the command line

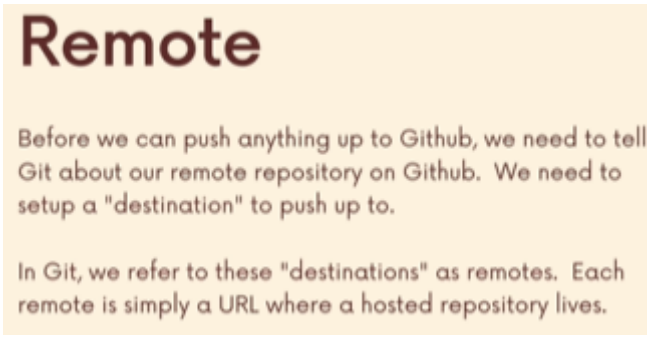
```
git remote add origin https://github.com/IDK9911/to_learn_git.git
git branch -M main
git push -u origin main
```

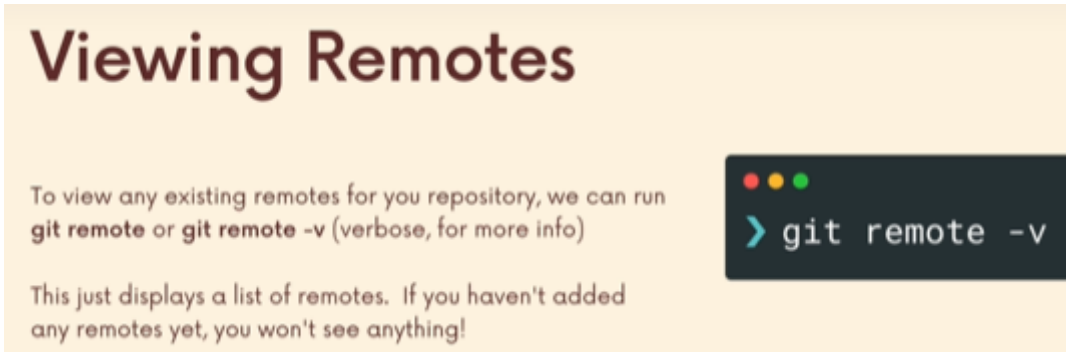
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

A Crash Course on Git Remotes



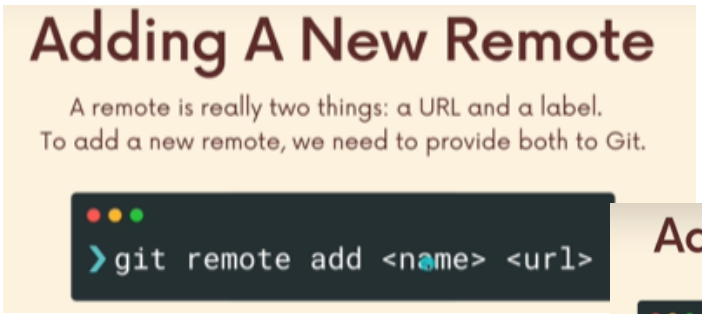


When i cloned a directory from github, i get this along with it.

```
2048 ❯ git remote                                                    master
origin
2048 ❯ git remote -v                                                 master
origin  https://github.com/gabrielecirulli/2048.git (fetch)
origin  https://github.com/gabrielecirulli/2048.git (push)
2048 ❯                                                               master
```

By setting this url (remote) , it gives us the ability to tell is there any new code on the url, is there any new updates.

Or if i was collaborator, hey git i would like to push up my 10 commits.



Origin?

Origin is a conventional Git remote name, but it is not at all special. It's just a name for a URL.

When we clone a Github repo, the default remote name setup for us is called origin. You can change it. Most people leave it.

Other commands

They are not commonly used, but there are commands to rename and delete remotes if needed.

git remote rename <old> <new>

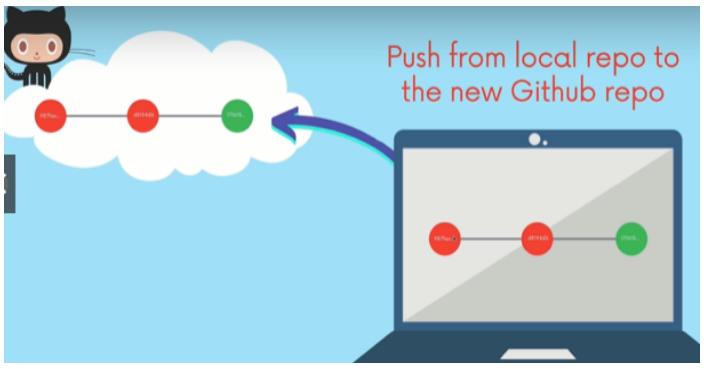
git remote remove <name>

Introducing Git Push

Option 1: Existing Repo

If you already have an existing repo locally that you want to get on Github...

- Create a new repo on Github
- Connect your local repo (add a remote)
- Push up your changes to Github



Pushing

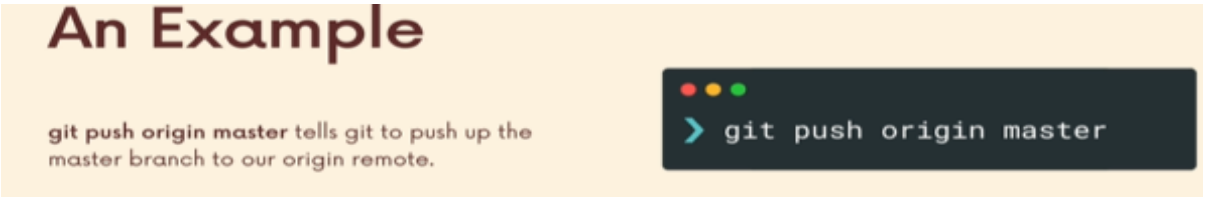
Now that we have a remote set up, let's push some work up to Github! To do this, we need to use the **git push** command.

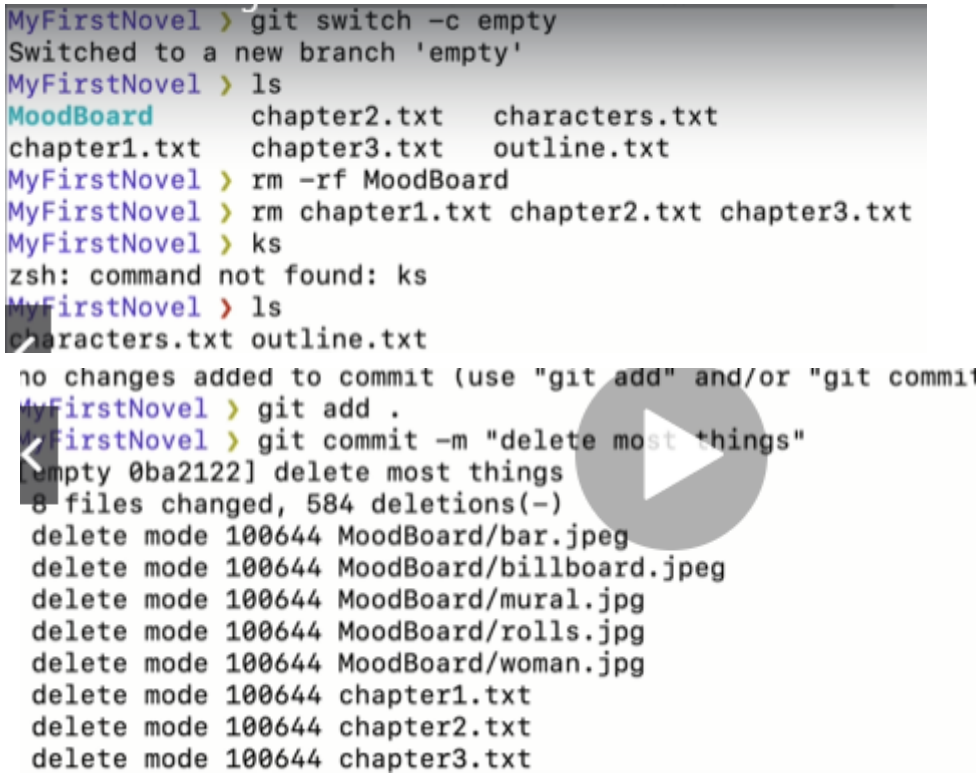
We need to specify the remote we want to push up to AND the specific local branch we want to push up to that remote.

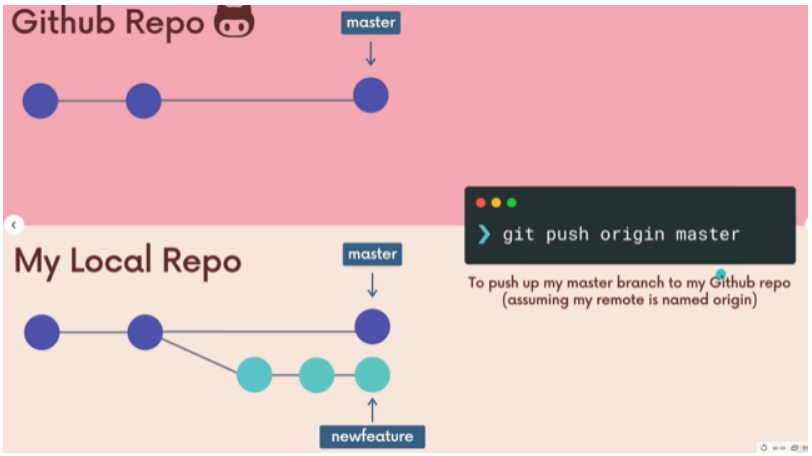
```
> git push <remote> <branch>
```

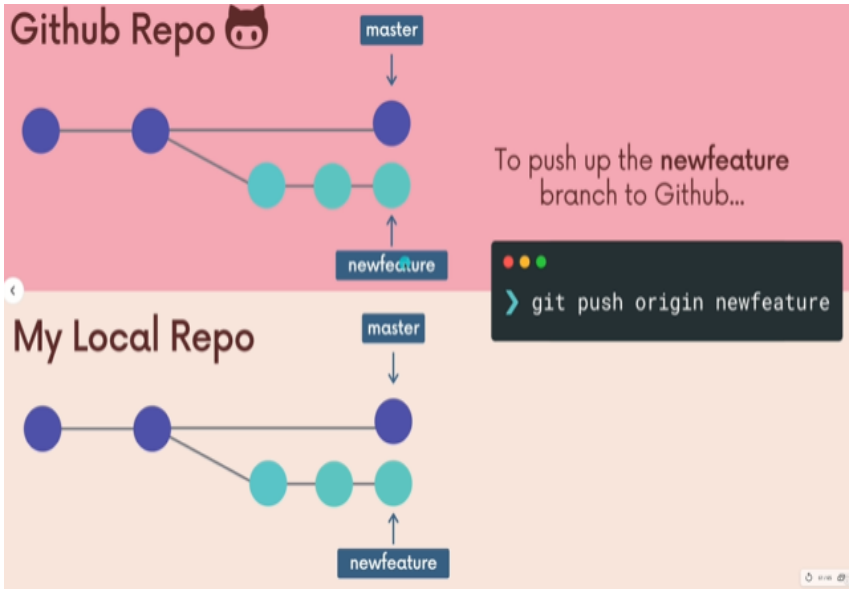
In a repository, we often have 7 or 8 branches, so when we push we may dont want to push them all on github.

Often it is master or main branch.

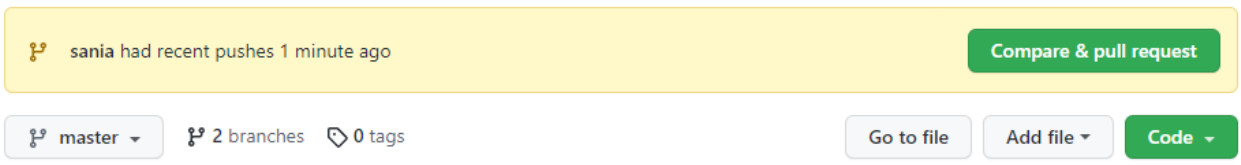


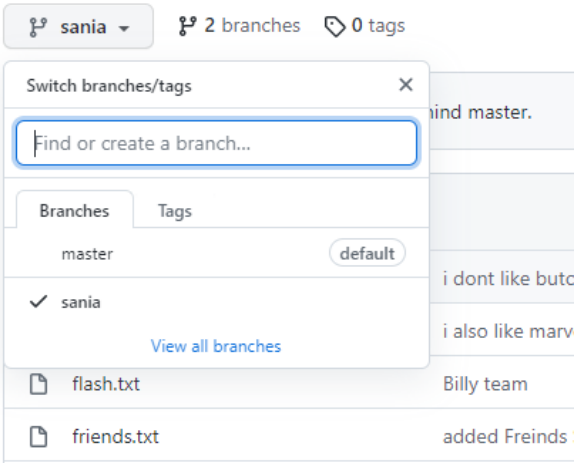







```
sahilsyed@DESKTOP-7VRCIK9:/mnt/c/Users/Dell/Desktop/Learning Git/kaju$ git push origin sania
Username for 'https://github.com': IDK9911
Password for 'https://IDK9911@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'sania' on GitHub by visiting:
remote:      https://github.com/IDK9911/to_learn_git/pull/new/sania
remote:
To https://github.com/IDK9911/to_learn_git.git
 * [new branch]      sania -> sania
```

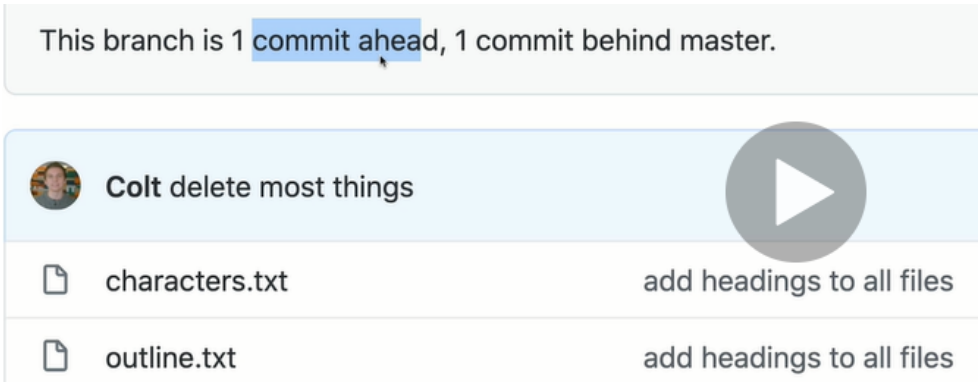




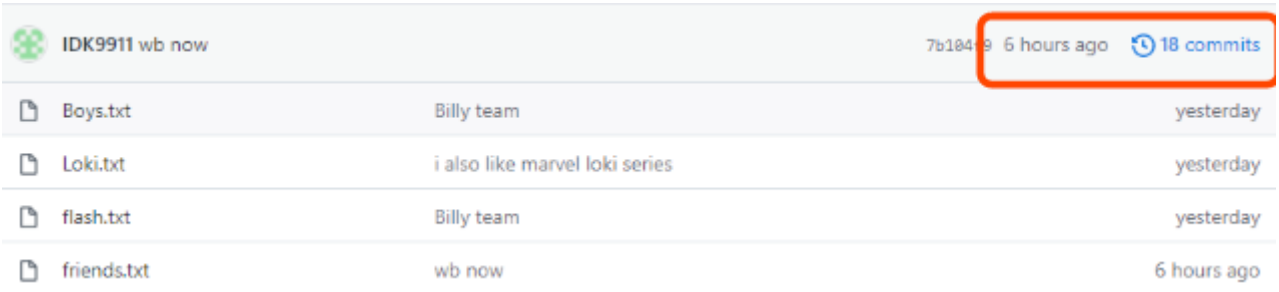

```
  (use "git add <file>..." to include in what wi
        chapter4.txt

nothing added to commit but untracked files prese
MyFirstNovel ❯ git add chapter4.txt
MyFirstNovel ❯ git commit -m "begin chapter 4"
[master 02c0f23] begin chapter 4
 1 file changed, 0 insertions(+), 0 deletions(-)
```

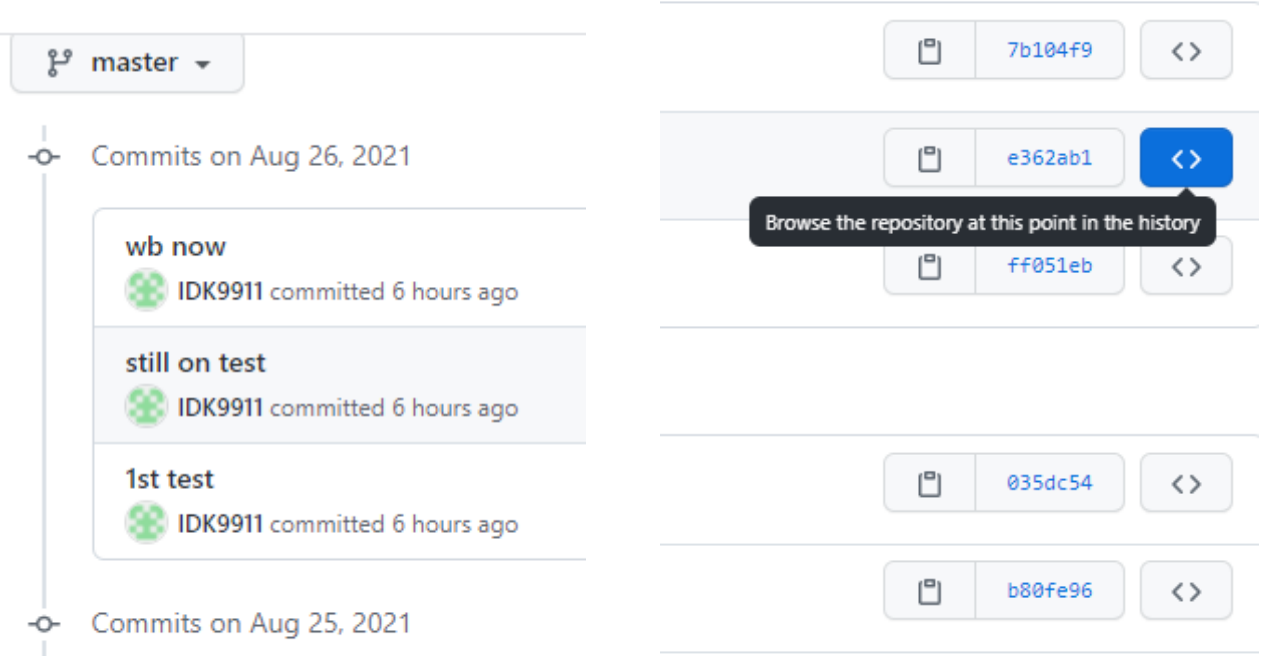
When we commit, we dont just push that commit we push the entire branch.

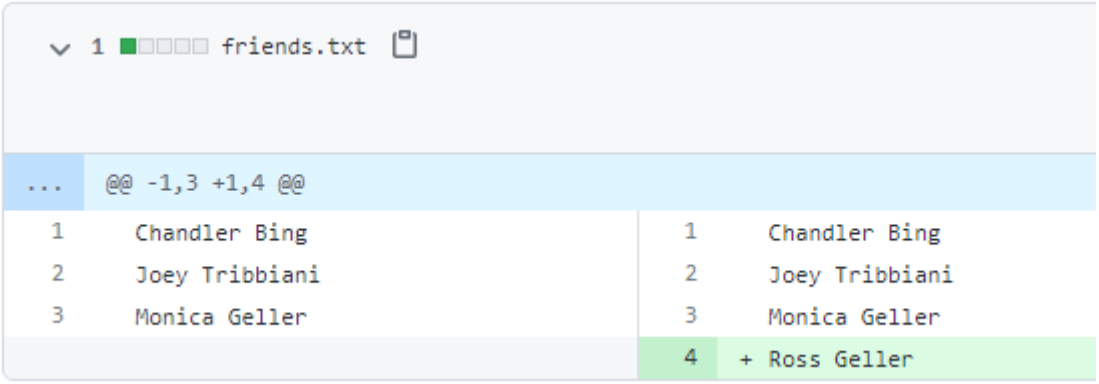
Touring a git repository



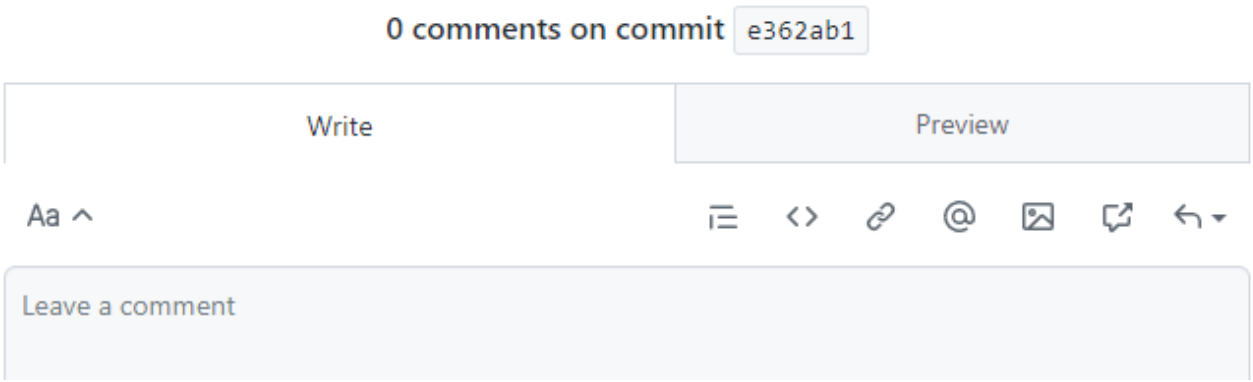
It means this branch has 1 commit that master doesnt have, and is missing One commit that master doesnt have.







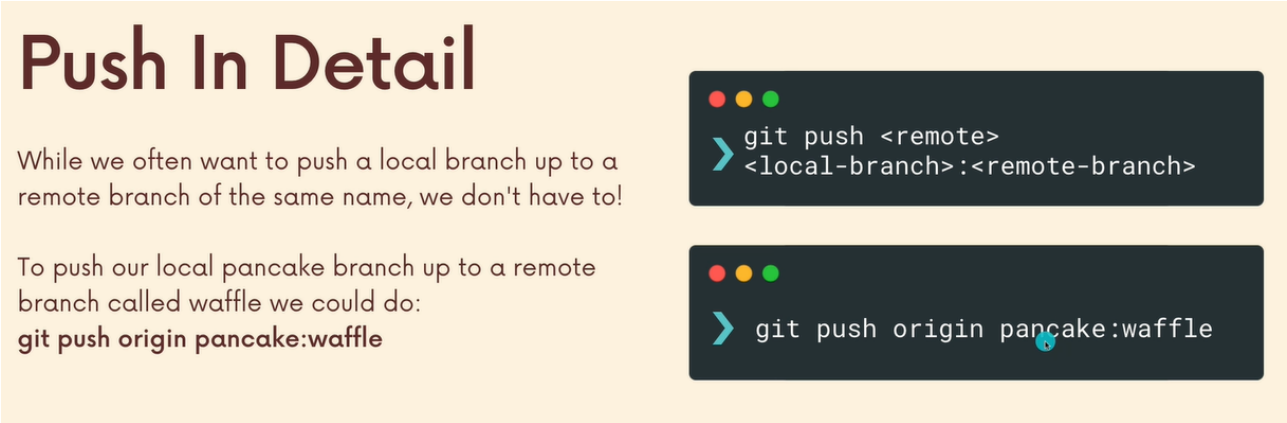
Note u can edit the file by clicking on the pencil icon. and then if u r on the main branch. u might not be able to do it.
Cuz of branch protection rules.
so u can create feature branch.



Commenting is really important. Its important for discussing changes as you start working with other people.

A closer Look At Git push


```
Pushing ❯ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 231 bytes | 231.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Colt/pushme.git
 * [new branch]      master -> master
```



Take the local branch pancake and push it to the origin branch waffle.

```
Pushing ❯ git push origin cats                                    cats
Pushing ❯ git push origin cats:master
```

```
Pushing ❯ git push origin cats:master                                cats
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Colt/pushme.git
   f508e11..f3d84fa  cats -> master
Pushing ❯                                                            cats
```

What does “git push -u” mean?

This tells git on my computer , i want to push up the master branch to origin and i want you to remember it.

```
Pushing ❯ git switch -c dogs                                    master
Switched to a new branch 'dogs'
Pushing ❯                                                         dogs
```

```
Pushing ❯ git push origin dogs                                    dogs
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dogs' on GitHub by visiting:
remote:      https://github.com/Colt/pushme/pull/new/dogs
remote:
To github.com:Colt/pushme.git
 * [new branch]      dogs -> dogs
Pushing ❯ touch dogs.txt                                          dogs
Pushing ❯ git add dogs.txt                                        dogs
Pushing ❯ git commit -m "create dogs"                             dogs
[dogs 7f0d5e2] create dogs
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dogs.txt
Pushing ❯ git push                                                dogs
```

```
Pushing ❯ git push
fatal: The current branch dogs has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin dogs
```

```
Pushing ❯ git push -u origin dogs
```

```
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Colt/pushme.git
   f508e11..7f0d5e2  dogs -> dogs
Branch 'dogs' set up to track remote branch 'dogs' fr
Pushing ❯ touch moredogs.txt
Pushing ❯ git add moredogs.txt
Pushing ❯ git commit -m "create more dogs"
[dogs 3910353] create more dogs
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 moredogs.txt
Pushing ❯ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
```

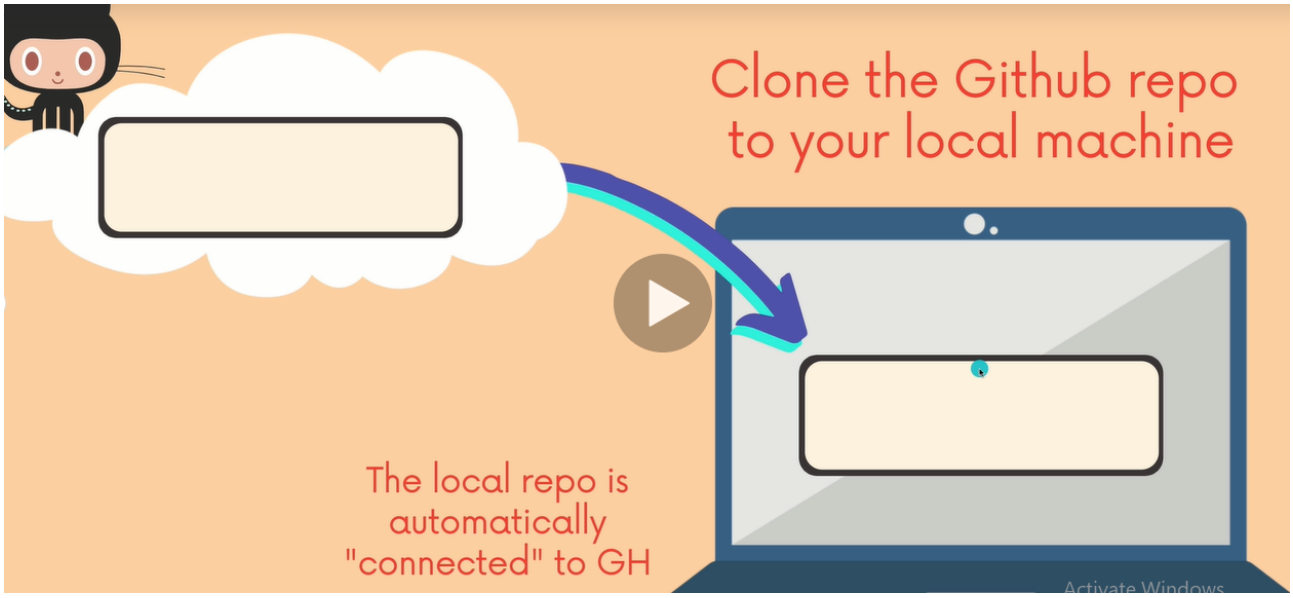
```
Pushing ❯ git push -u origin dogs:cats
```

If i ran the above, not only it would make dog branch at cats on github. But it will remember this too.

git push -u origin cats.

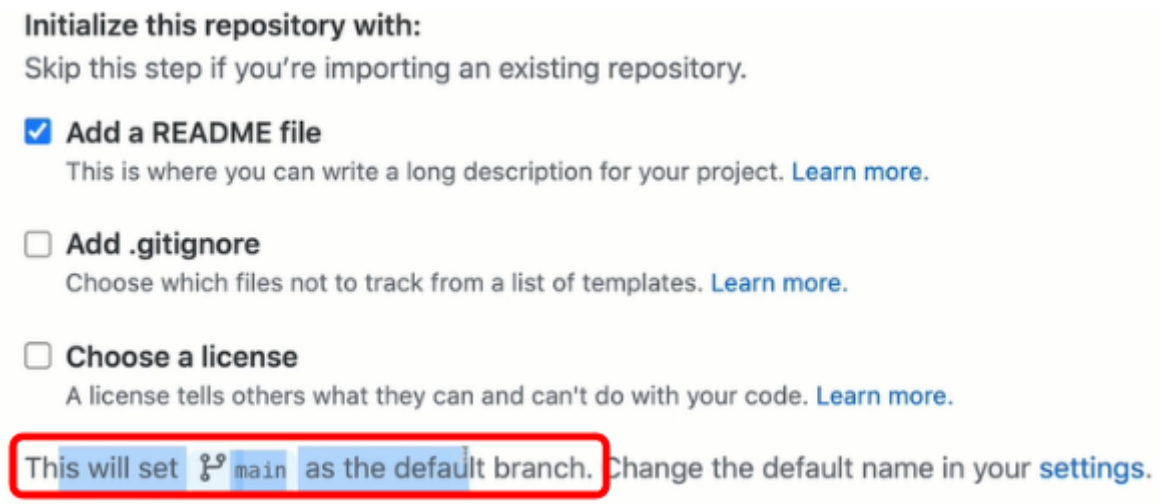
So be careful.

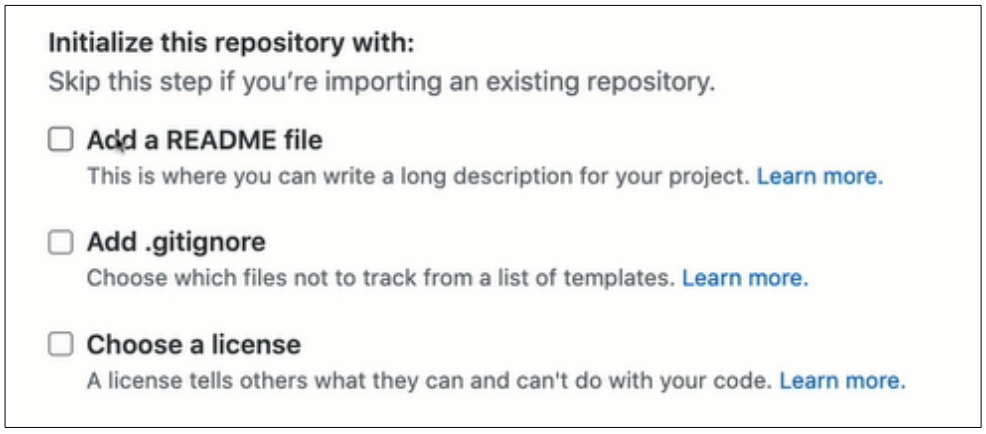
Another Github Workflow: Cloning first.



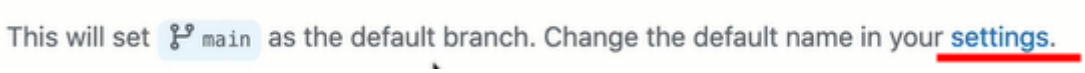
Automically do the remote for us.

Main & Master : Github Default branches





if i uncheck Readme, it doesnt say anything about Main, cuz there is no default branch.



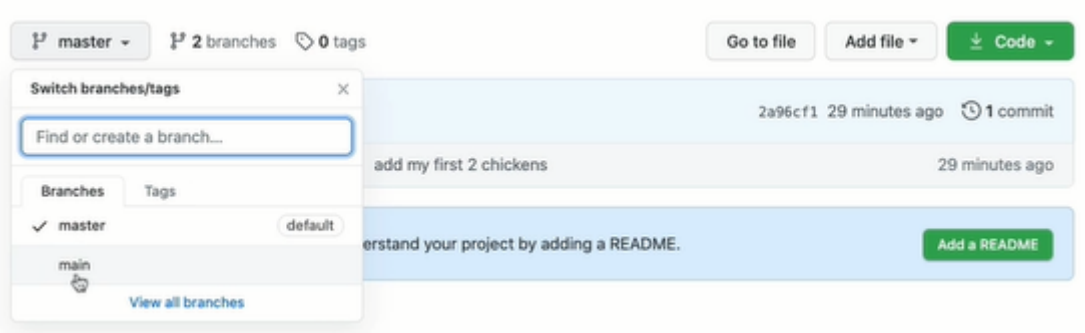
We can change that in our settings that it always default to master.


```
git remote add origin https://github.com/Colt/asdasd.git
git branch -M main
git push -u origin main
```

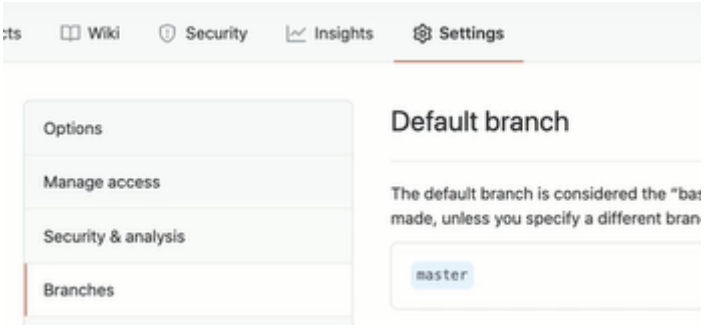
< it tells to rename the branch we currently on with main

```
chickens-demo ❯ git branch                                    master
chickens-demo ❯ git branch -M main                            master
chickens-demo ❯                                                 main
```

```
chickens-demo ❯ git push origin main
Total 0 (delta 0), reused 0 (delta 0)
```



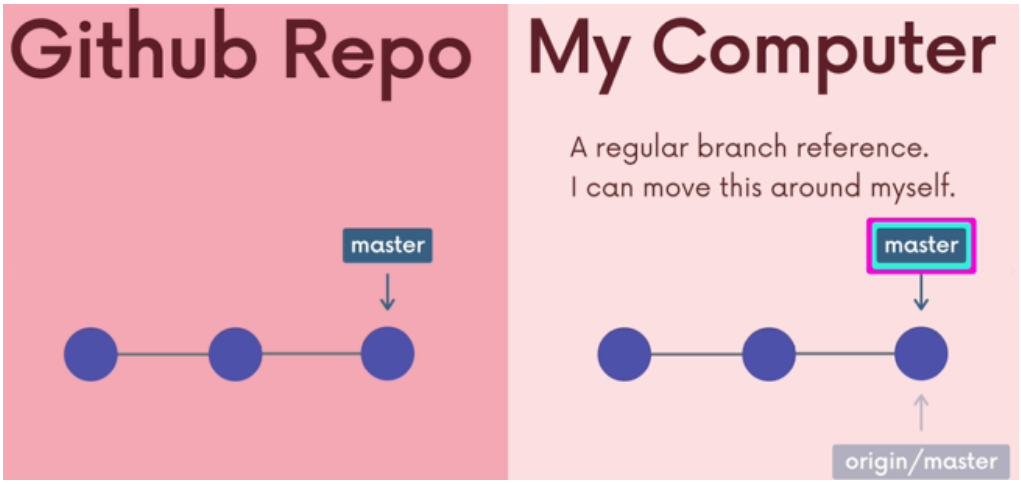
You will Still see master is still consider the default branch. Bcoz we push master first.



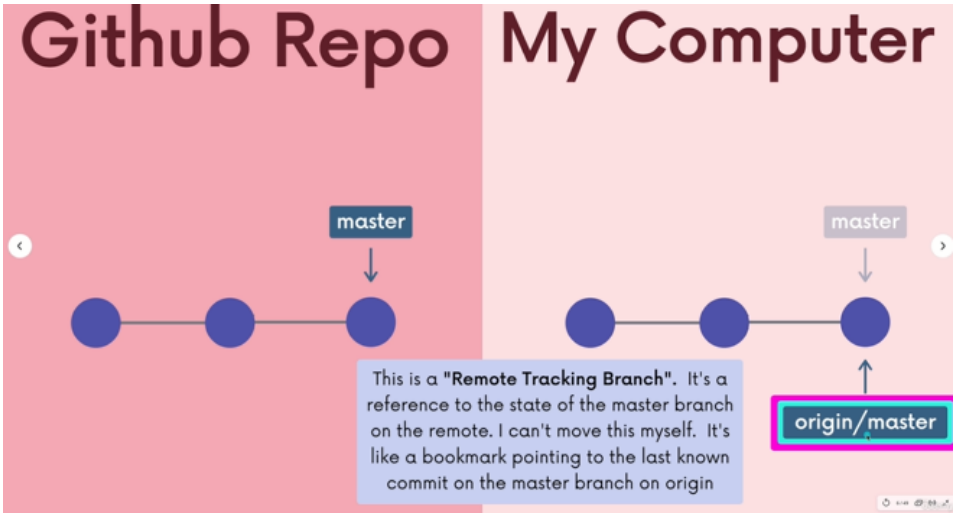
Fetching and Pulling

We are talking about remote repositories, we are talking about git fetch and git pull. They both are talking about getting changes down from the remote.

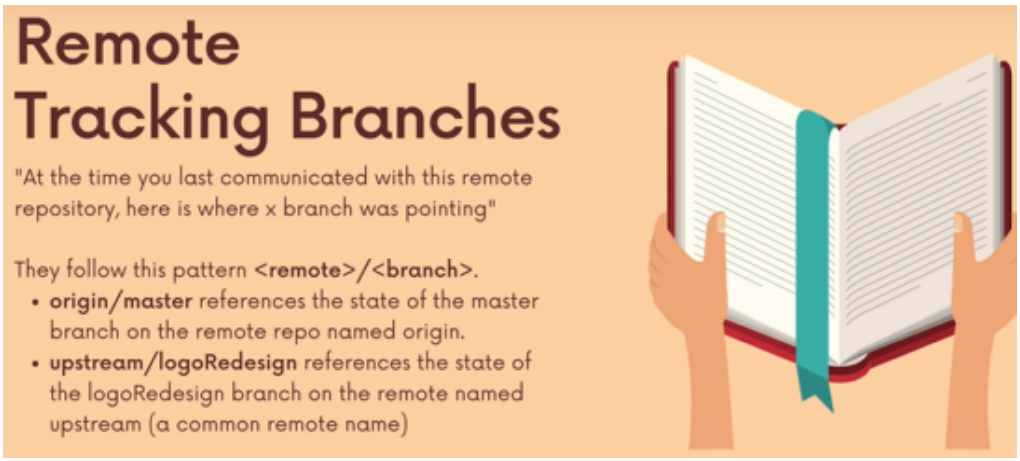
Remoting tracking branches : WTF are they



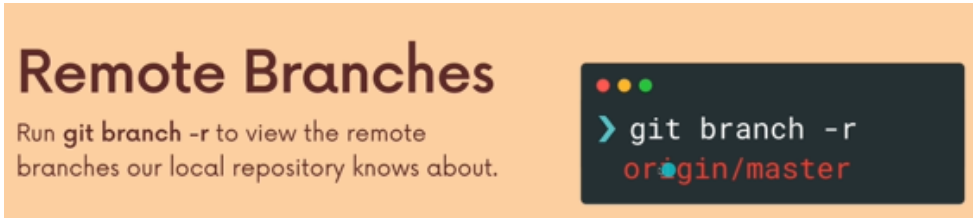
Whatever the github repo as its defualt repository, we gonna have that, in this e.g its master, we gonna start with that as our default.

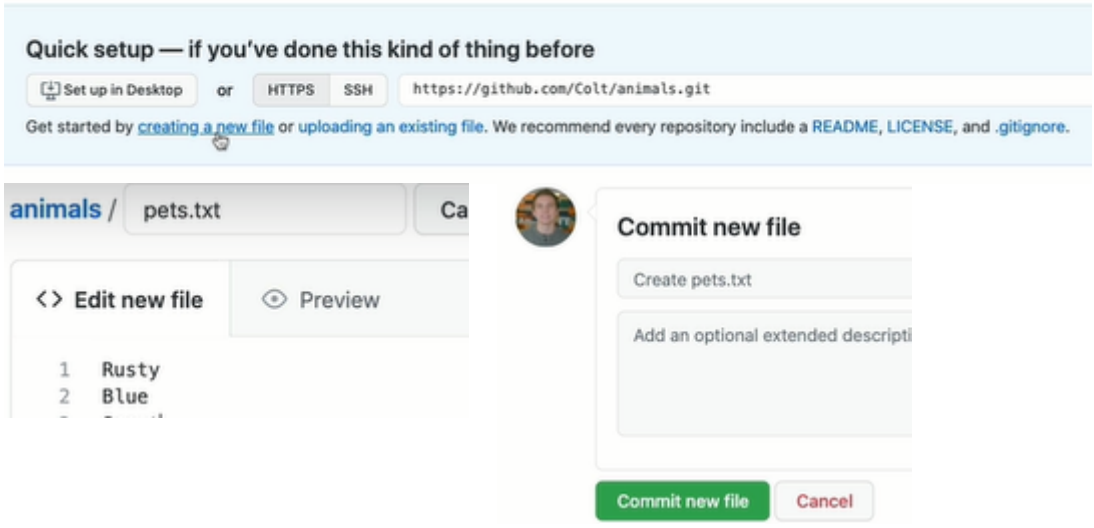


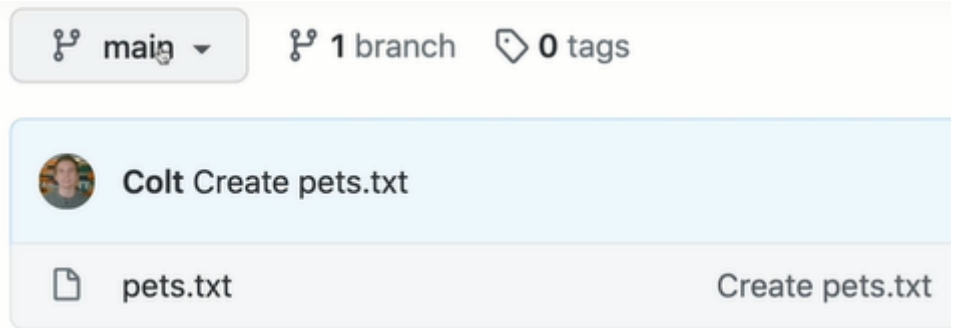
master and origin/master may start form the same starting point, but they might diverge as we do more work aka commits in our local repository.



Think of remote tracking branch as the last time you communicated with this remote repository. Here is where, the master branch, pointing at the github repository.







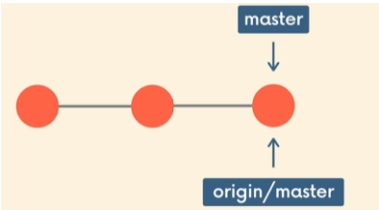
```
animals ❯ git remote -v
origin  https://github.com/Colt/animals.git (fetch)
origin  https://github.com/Colt/animals.git (push)
```

```
animals ❯ git remote -v                                                    main
origin  https://github.com/Colt/animals.git (fetch)
origin  https://github.com/Colt/animals.git (push)
```

above, git setup this remote tracking branch called origin at Main.

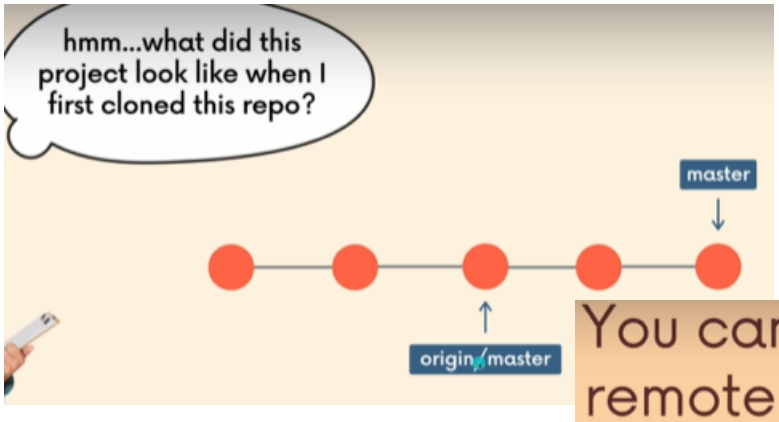
```
animals ❯ git branch -r
  origin/HEAD -> origin/main
  origin/main
(END)
```

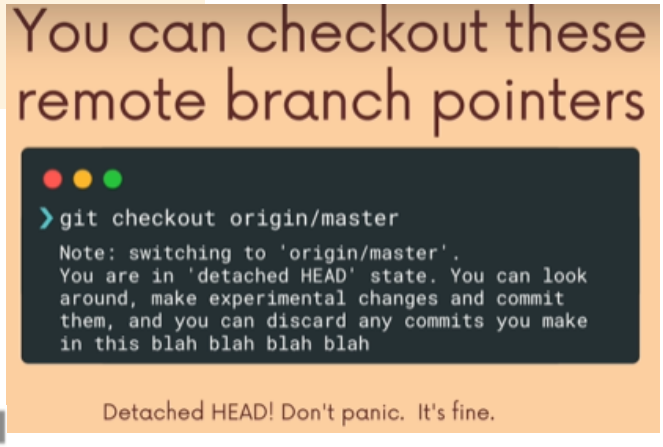
Checking out remote tracking branch



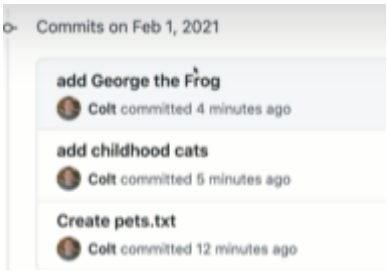

```
animals ❯ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
animals ❯ git add pets.txt
animals ❯ git commit -m "add George the Frog"
```



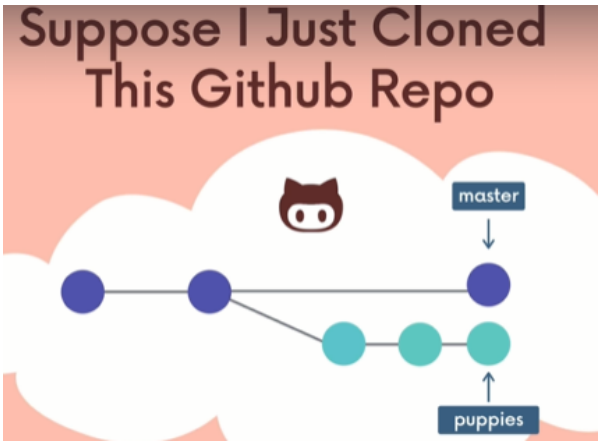


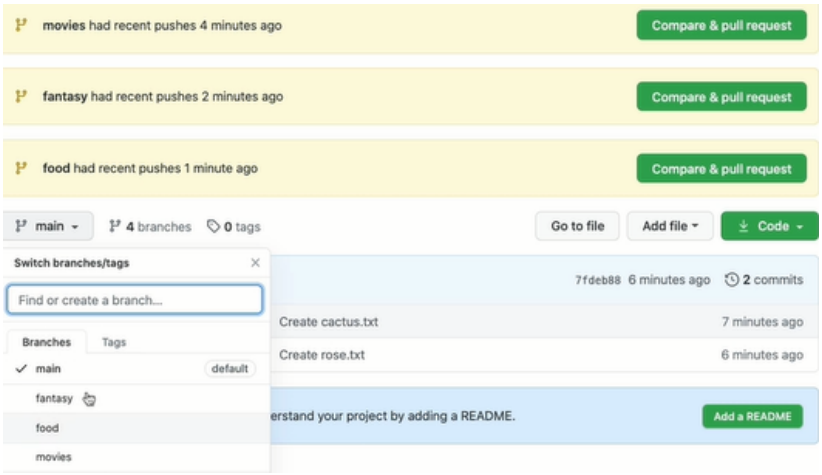

```
animals ❯ git checkout origin/main
animals ❯ git push origin main
```

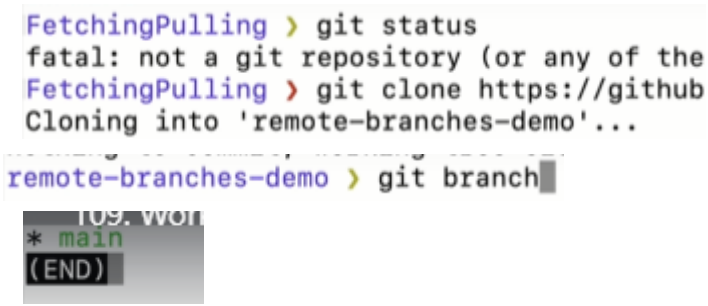


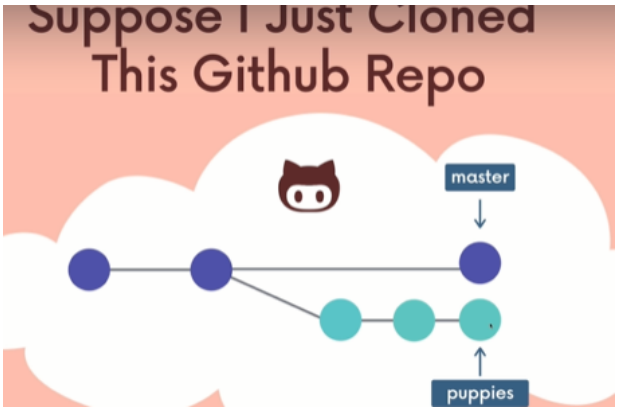

```
animals ❯ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
animals ❯
```

Working With Remote Branches









Remote Branches

Once you've cloned a repository, we have all the data and Git history for the project at that moment in time. However, that does not mean it's all in my workspace!

The github repo has a branch called **puppies**, but when I run **git branch** I don't see it on my machine! All I see is the master branch. What's going on?

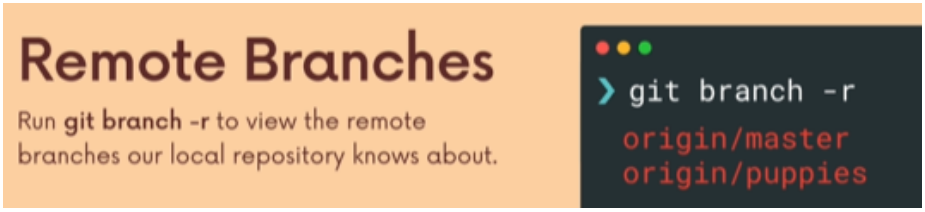
```
❯ git branch
*master
```

```
remote-branches-demo ❯ git branch -r
  origin/HEAD -> origin/main
  origin/fantasy
  origin/food
  origin/main
  origin/movies
(END)
```

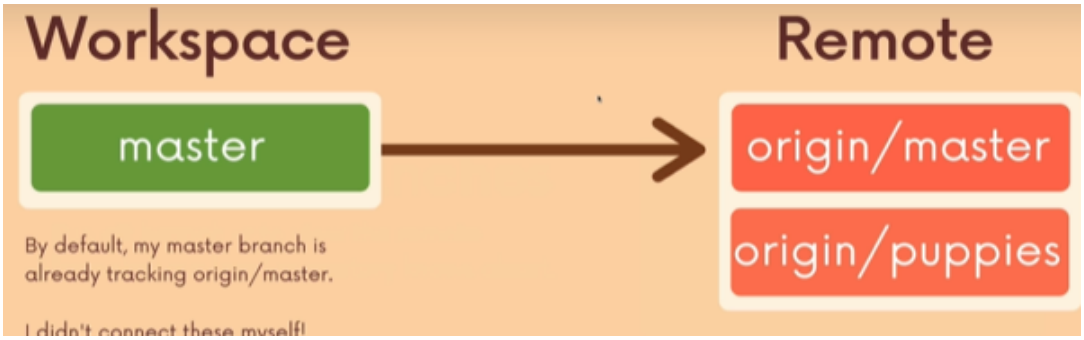
We see everyone of the branches, it has a remote tracking reference. It can tell me where each one his. Just like a bookmark.

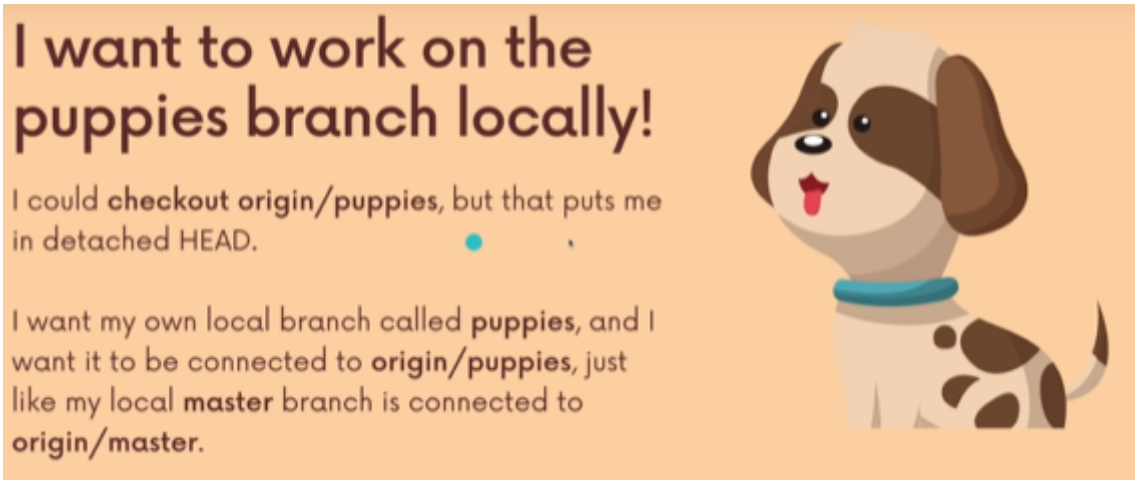
Below, but we don't see the branches.

```
* main
(END)
```



Whenever we clone,



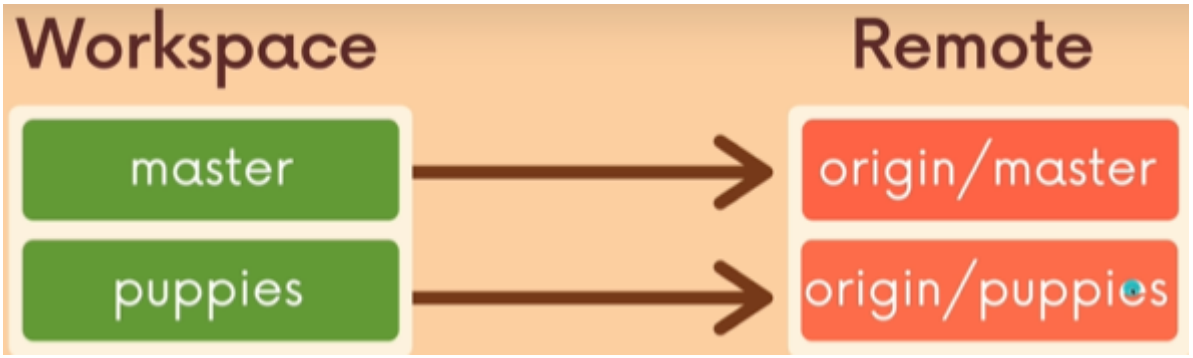


```
remote-branches-demo ❯ git checkout origin/food
Note: switching to 'origin/food'.

You are in 'detached HEAD' state. You can look around,
```

```
remote-branches-demo ❯ ls
banana.txt   popsicle.txt
```

HEAD

So if we switch to a branch with that exact name, so if git detects that we are switching to a branch puppies which doesn't exist. It will check then remote branches in the repositories and will make the connection.

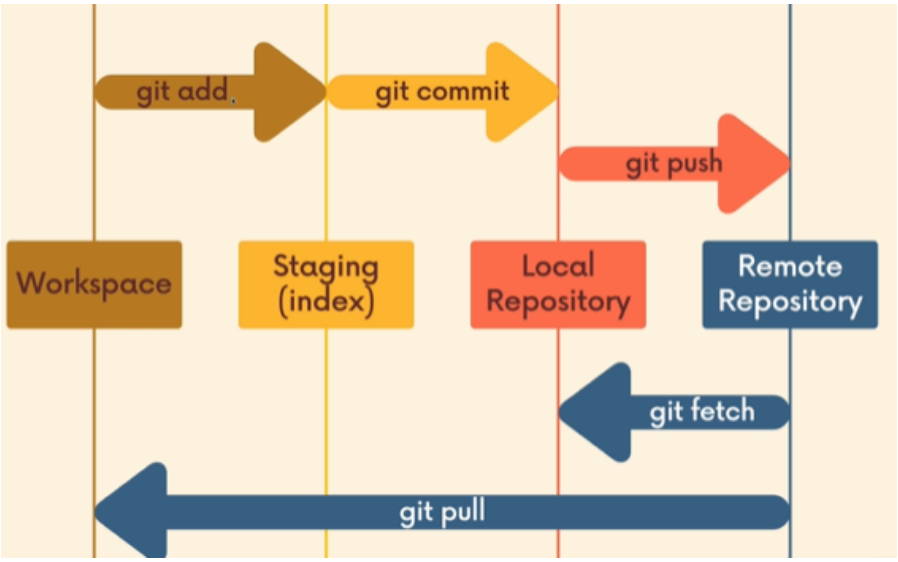


```
origin/HEAD -> origin/main
origin/fantasy
origin/food
origin/main
origin/movies
```

```
remote-branches-demo ❯ git switch movies                                    main
Branch 'movies' set up to track remote branch 'movies' from 'origin'.
Switched to a new branch 'movies'
```


```
remote-branches-demo ❯ git status                                           fantasy
On branch fantasy
Your branch is ahead of 'origin/fantasy' by 1 commit.
  (use "git push" to publish your local commits)
```

Git Fetch: The Basics



Fetching

Fetching allows us to download changes from a remote repository, BUT those changes will not be automatically integrated into our working files.

It lets you see what others have been working on, without having to merge those changes into your local repo.

Think of it as "please go and get the latest information from Github, but don't screw up my working directory."

Git Fetch

The **git fetch <remote>** command fetches branches and history from a specific remote repository. It only updates remote tracking branches.

git fetch origin would fetch all changes from the origin remote repository.

```
❯ git fetch <remote>
```

If not specified, <remote> defaults to origin

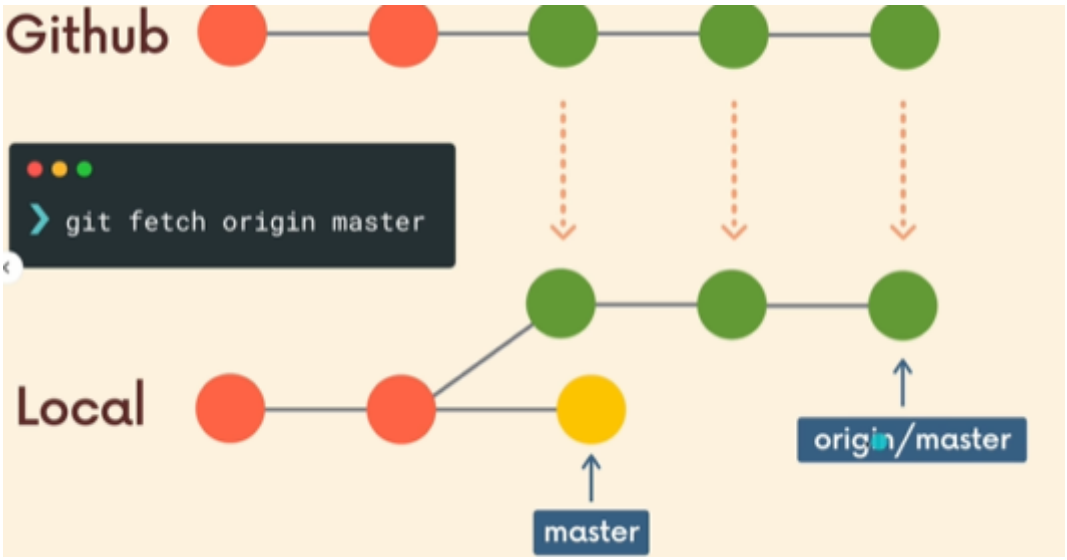
Git Fetch

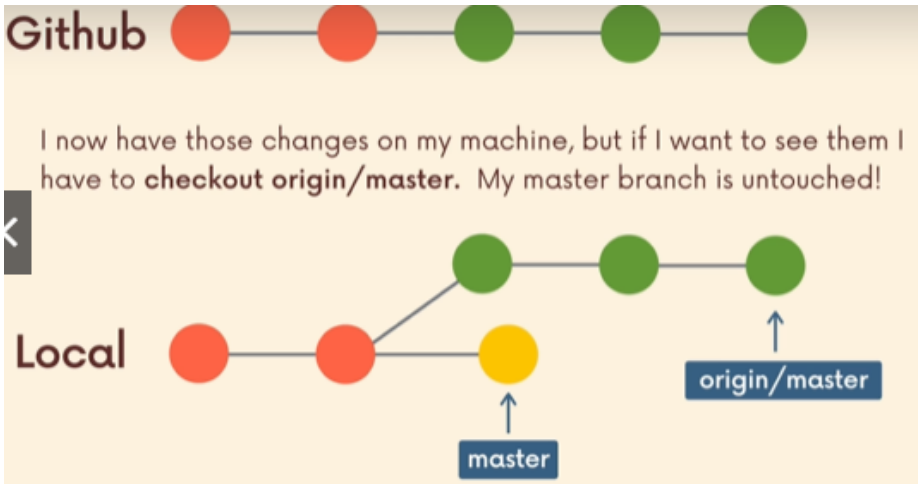
We can also fetch a specific branch from a remote using **git fetch <remote> <branch>**

For example, **git fetch origin master** would retrieve the latest information from the master branch on the origin remote repository.

```
❯ git fetch <remote> <branch>
```

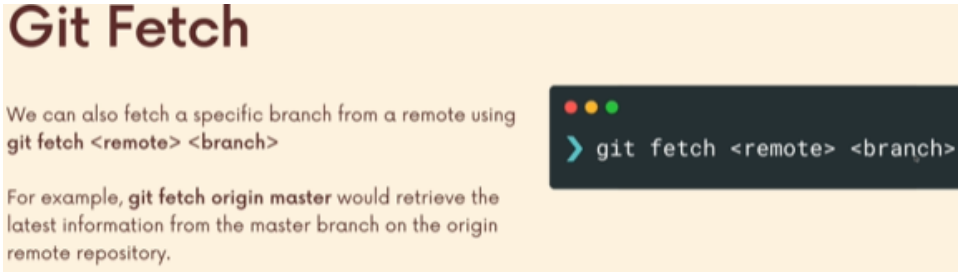

Github
Uh oh! The remote repo has changed! A teammate has pushed up changes to the master branch, but my local repo doesn't know!
How do I get those changes???
Local
origin/master
master


Github
git fetch origin master
Local
origin/master
master


Github
I now have those changes on my machine, but if I want to see them I have to checkout origin/master. My master branch is untouched!
Local
origin/master
master

Demonstrating Git Fetch

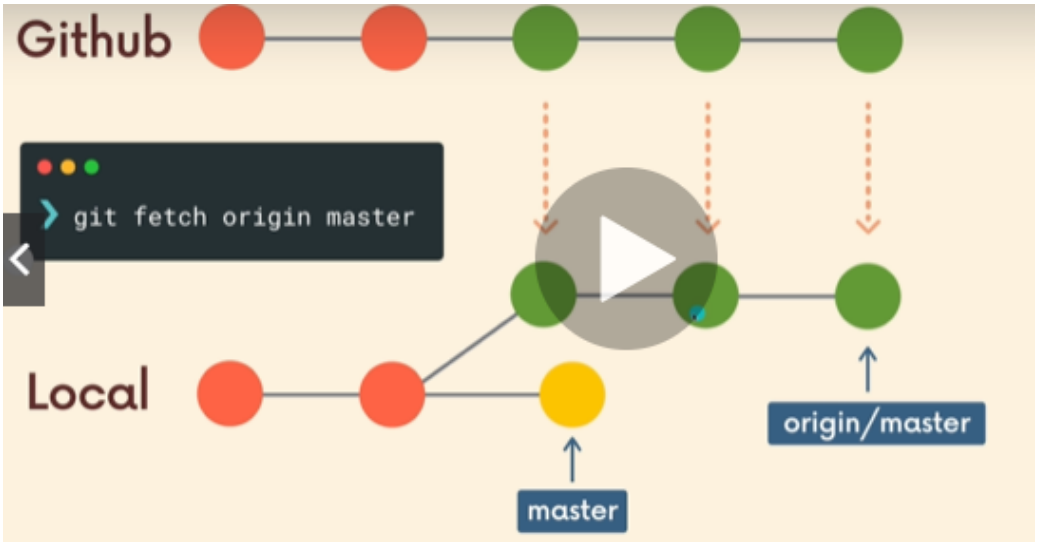

```
remote-branches-demo ❯ git status                                    movies
On branch movies
Your branch is up to date with 'origin/movies'.

nothing to commit, working tree clean
```




```
remote-branches-demo ❯ git fetch origin                              movies
remote-branches-demo ❯                                               movies
```

```
remote-branches-demo ❯ git status                                    movies
On branch movies
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
remote-branches-demo ❯                                               movies
```

```
remote-branches-demo ❯ git status                                    movies
On branch movies
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
remote-branches-demo ❯                                               movies
```

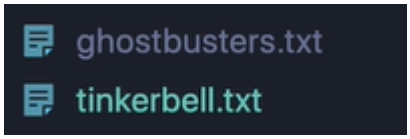



```
remote-branches-demo ❯ git log
```

```
commit 6c32e31907895ebbf9a0df486b1afbd9a3952e7 (HEAD -> movies
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 14:05:17 2021 -0800

    Create bambi.txt

commit 335cbf5f7edb25a79cb666a93ee464a461089f4e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 14:04:48 2021 -0800

    Create ghostbusters.txt

commit 2cfeb8a1982ad1f684a93b945ca95eacb901c8f0
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 14:04:32 2021 -0800

    Delete rose.txt
```

You don’t see anything new.

```
remote-branches-demo ❯ git checkout origin/movies
```

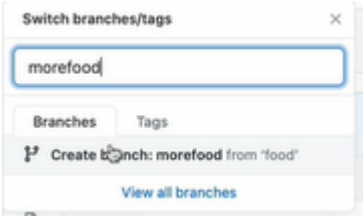


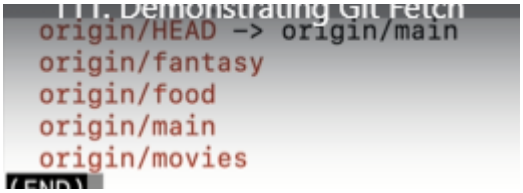

```
commit d94a89ed4af3475ecb89634c0aa31c44f4b2836 (HEAD, origin/movies)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 16:42:21 2021 -0800

    Create tinkerbell.txt

commit 6c32e319078957ebbf9a0df486b1afbd9a3952e7 (movies)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 14:05:17 2021 -0800

    Create bambi.txt

commit 335cbf5f7edb25a79cb666a93ee464a4b1089f4e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 14:04:48 2021 -0800

    Create ghostbusters.txt
```

```
remote-branches-demo ❯ git fetch origin food
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/Colt/remote-branches-demo
 * branch            food       -> FETCH_HEAD
   ccda9ed..c4f3cb7  food       -> origin/food
```

```
remote-branches-demo ❯ git checkout origin/food
Note: switching to 'origin/food'.

You are in 'detached HEAD' state. You can look around, m
changes and commit them, and you can discard any commits
```

Git fetch is not just about updating the particular branch that you are on and getting the latest. Also it might be that new branch was created.





I don’t see morefood.

```
remote-branches-demo ❯ git fetch
From https://github.com/Colt/remote-branches-demo
 * [new branch]      morefood   -> origin/morefood
```

Git Pull

Pulling

git pull is another command we can use to retrieve changes from a remote repository. Unlike fetch, pull actually updates our HEAD branch with whatever changes are retrieved from the remote.

"go and download data from Github AND immediately update my local repo with those changes"

git pull = git fetch + git merge

update the remote tracking branch with the latest changes from the remote repository

update my current branch with whatever changes are on the remote tracking branch

git pull

To pull, we specify the particular remote and branch we want to pull using **git pull <remote> <branch>**. Just like with git merge, it matters WHERE we run this command from. Whatever branch we run it from is where the changes will be merged into.

git pull origin master would fetch the latest information from the origin's master branch and merge those changes into our current branch.

```
❯ git pull <remote> <branch>
```

pulls can result in merge conflicts!!

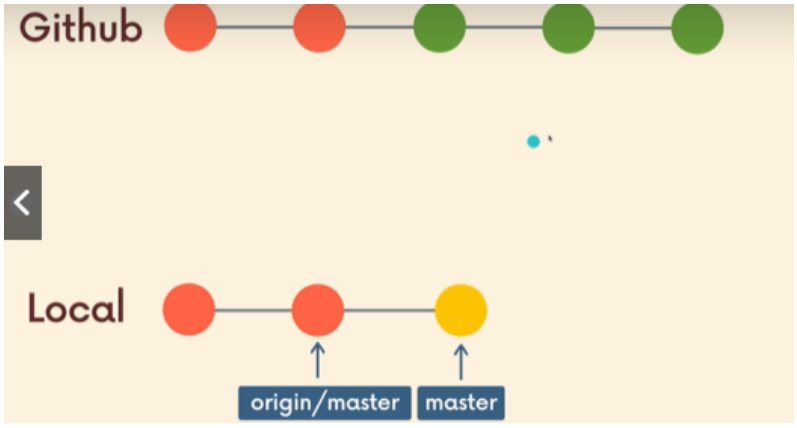
```
remote-branches-demo ❯ git pull origin movies                    movies
From https://github.com/Colt/remote-branches-demo
 * branch            movies     -> FETCH_HEAD
Updating 6c32e31..d94a89e
Fast-forward
 tinkerbell.txt | 37 ++++++++++++++++++++++++++++++++++++
 1 file changed, 37 insertions(+)
 create mode 100644 tinkerbell.txt
```

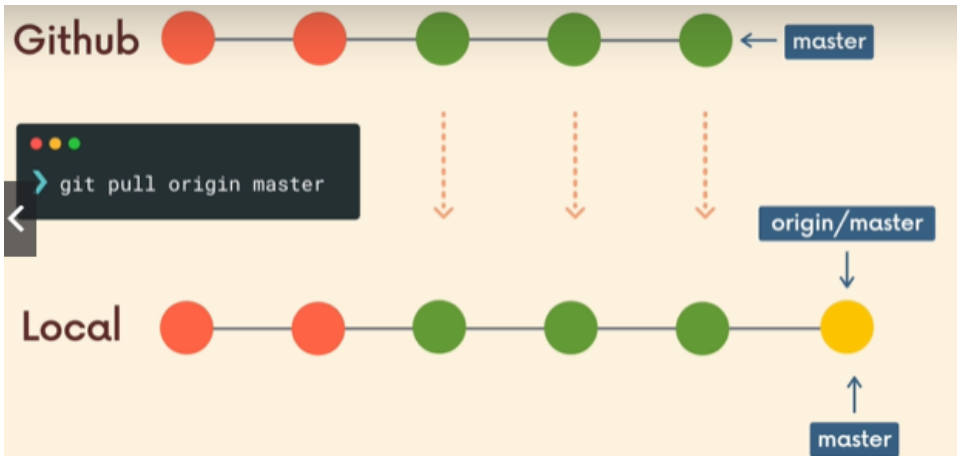
```
commit d94a89ed4af37475ecb89634c0aa31c44f4b2836 (HEAD -> movies, origin/mov
ies)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 16:42:21 2021 -0800

    Create tinkerbell.txt
```

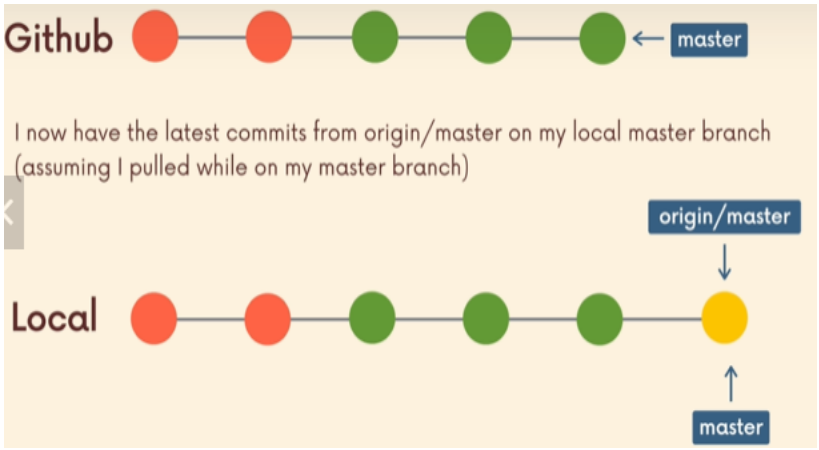
Git Pull & Merge Conflicts

pulls can result in merge conflicts!!

Sometimes you may have some work locally, that is not on github and github might have some commits that you don't have locally. So when you pulled down there is gonna be conflicts.



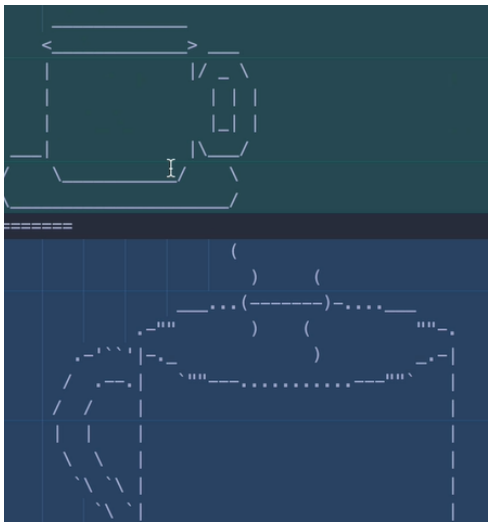





```
        coffee.txt

nothing added to commit but untracked files present (use "git add" to tra
)
remote-branches-demo ❯ git add coffee.txt                              food
remote-branches-demo ❯ git commit -m "add my coffee art"               food
```

Good practice, before you push anything on github, you might wanna pull down and see if theres any change.

```
remote-branches-demo ❯ git pull origin food                            food
From https://github.com/Colt/remote-branches-demo
 * branch            food       -> FETCH_HEAD
CONFLICT (add/add): Merge conflict in coffee.txt
Auto-merging coffee.txt
Automatic merge failed; fix conflicts and then commit the result.
remote-branches-demo ❯                                                 food
```



```
remote-branches-demo ❯ git commit -m "fix merge conflicts
[food 8690895] fix merge conflicts
remote-branches-demo ❯ git status
On branch food
Your branch is ahead of 'origin/food' by 2 commits.
  (use "git push" to publish your local commits)
```

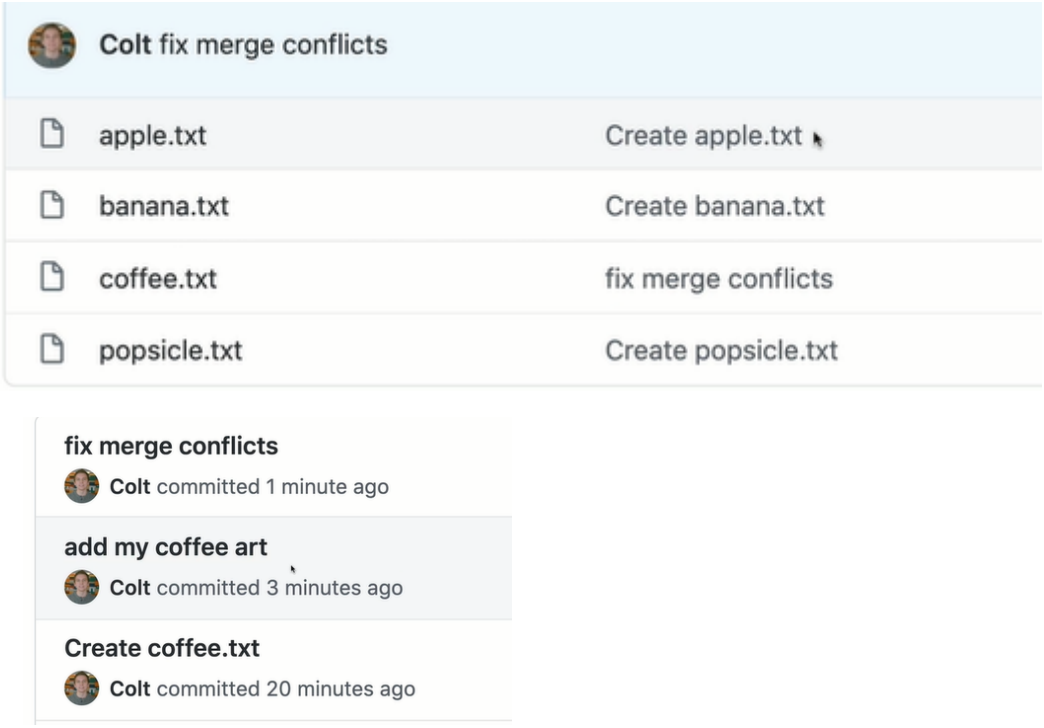
```
remote-branches-demo ❯ git status
On branch food
Your branch is ahead of 'origin/food' by 2 commits.
  (use "git push" to publish your local commits)
```

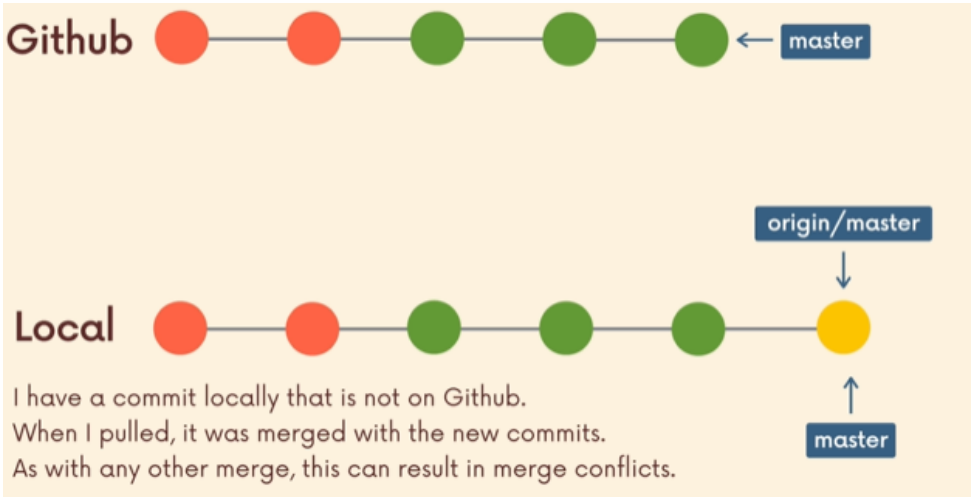
```
113. Git Pull & Merge Conflicts
commit 86908959ace692e7ef34f84f6a8995b31729ed4c (HEAD -> food)
Merge: dece76e c4f3cb7
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 17:07:44 2021 -0800

    fix merge conflicts

commit dece76e13c9c99be9f5be158be216be1edbb656d
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date:   Mon Feb 1 17:06:10 2021 -0800

    add my coffee art
```

The reason its ahead, the add my coffee art was my first commit(which was local).

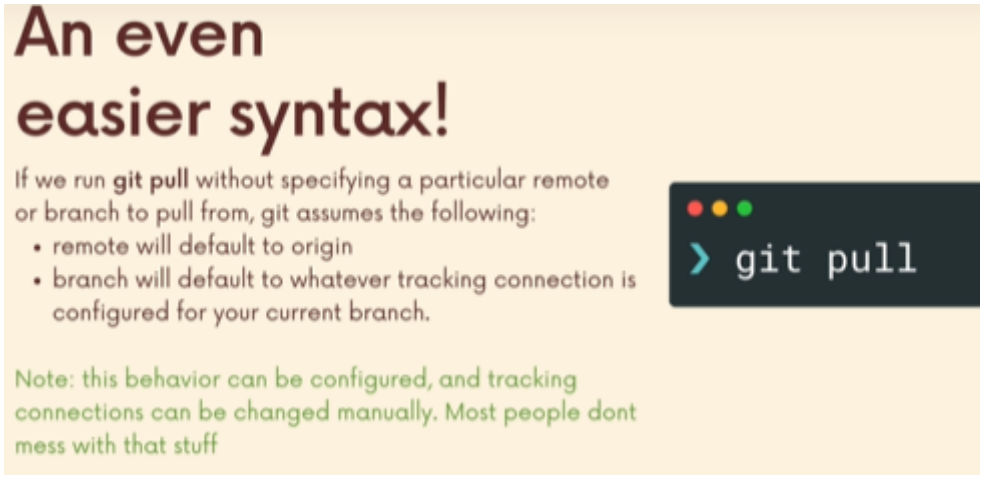
The fix merge conflicts when we pull the github commits and merge them with our locally committed changes.

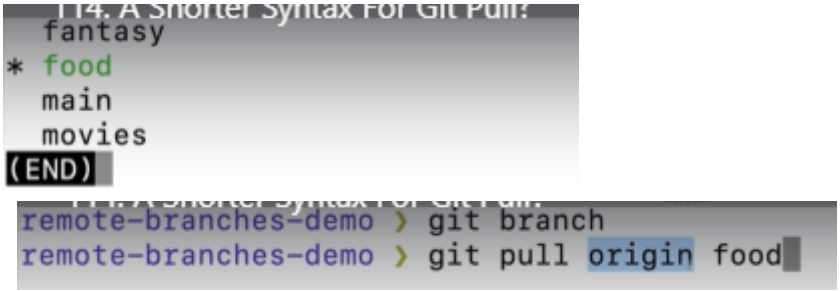
```
remote-branches-demo ❯ git push origin food
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
```



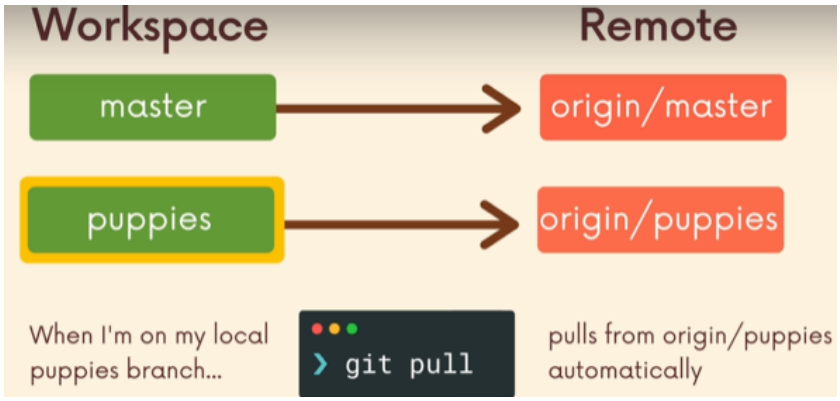


A shorter Syntax For Git Pull





We never setup the tracking connection.



```
remote-branches-demo ❯ git branch                                    food
remote-branches-demo ❯ git pull                                      food
Updating 8690895..ae2d02f
Fast-forward
 tea.txt | 20 ++++++++++++++++++++
 1 file changed, 20 insertions(+)
 create mode 100644 tea.txt
remote-branches-demo ❯ ls                                            food
apple.txt    banana.txt   coffee.txt   popsicle.txt tea.txt
remote-branches-demo ❯ git log                                       food
remote-branches-demo ❯ git branch                                    food
remote-branches-demo ❯ git switch fantasy                            food
Switched to branch 'fantasy'
Your branch is behind 'origin/fantasy' by 1 commit, and can be fast-forwar
ed.
  (use "git pull" to update your local branch)
```

```
remote-branches-demo ❯ git pull
Updating 463f23f..edd407b
Fast-forward
 gryphon.txt | 20 ++++++++++++++++++++
 1 file changed, 20 insertions(+)
 create mode 100644 gryphon.txt
```

git fetch

- Gets changes from remote branch(es)
- Updates the remote-tracking branches with the new changes
- Does not merge changes onto your current HEAD branch
- Safe to do at anytime

git pull

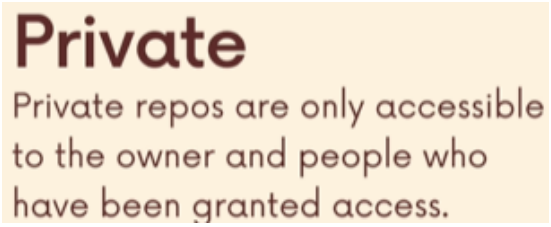
- Gets changes from remote branch(es)
- Updates the current branch with the new changes, merging them in
- Can result in merge conflicts
- Not recommended if you have uncommitted changes!

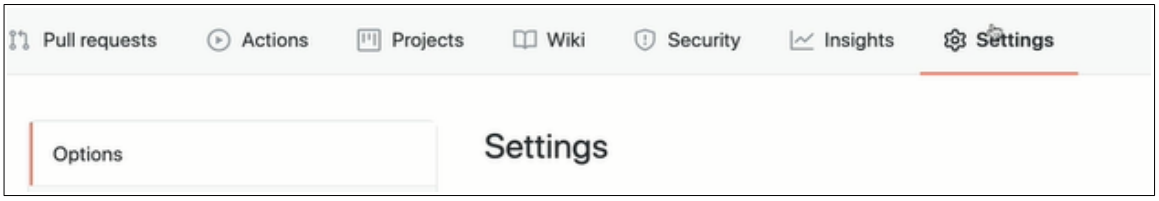
Github Grab bags Odds and ends

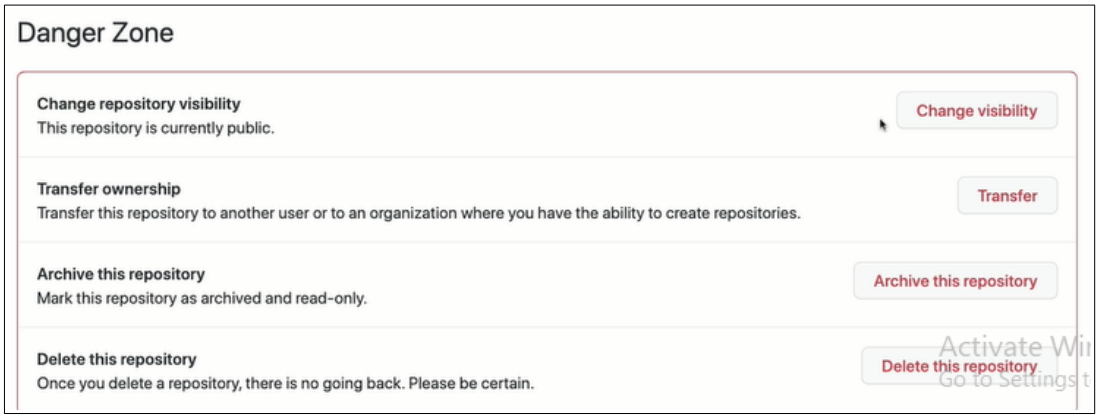
Readme.md , Md stands for mark down.

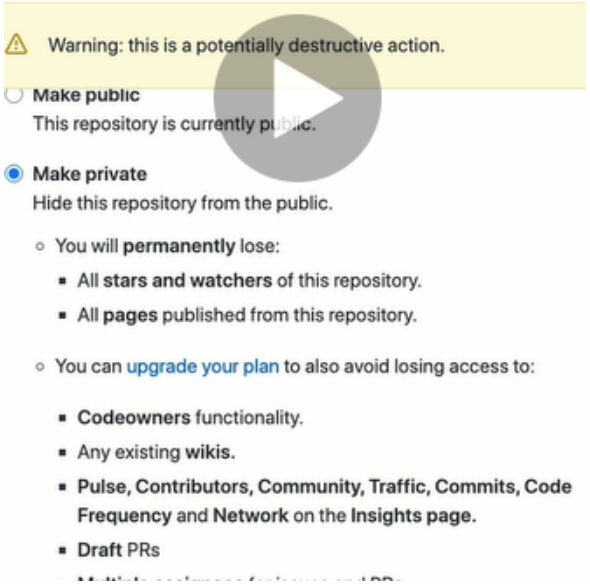
Github Pages, a very nice tool from Github that allows us it helps us to easily host static web pages from our repositories.

Github Repo Visibility: Public vs Private

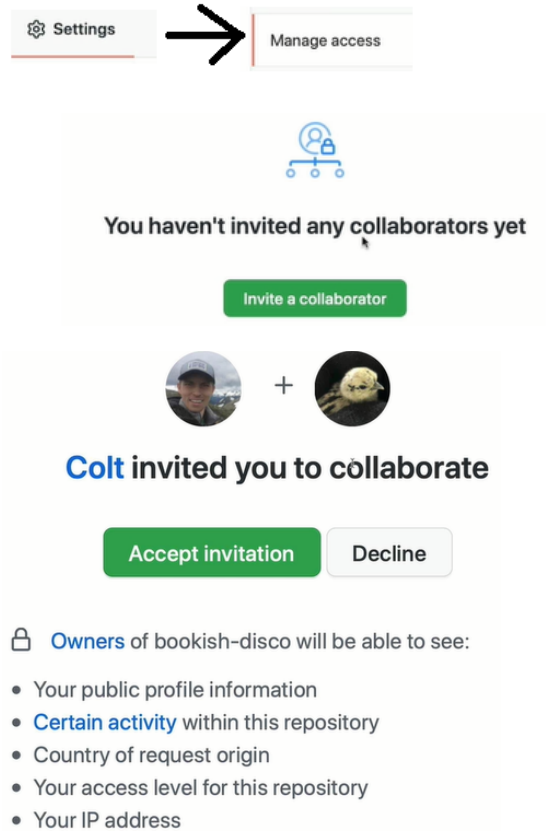




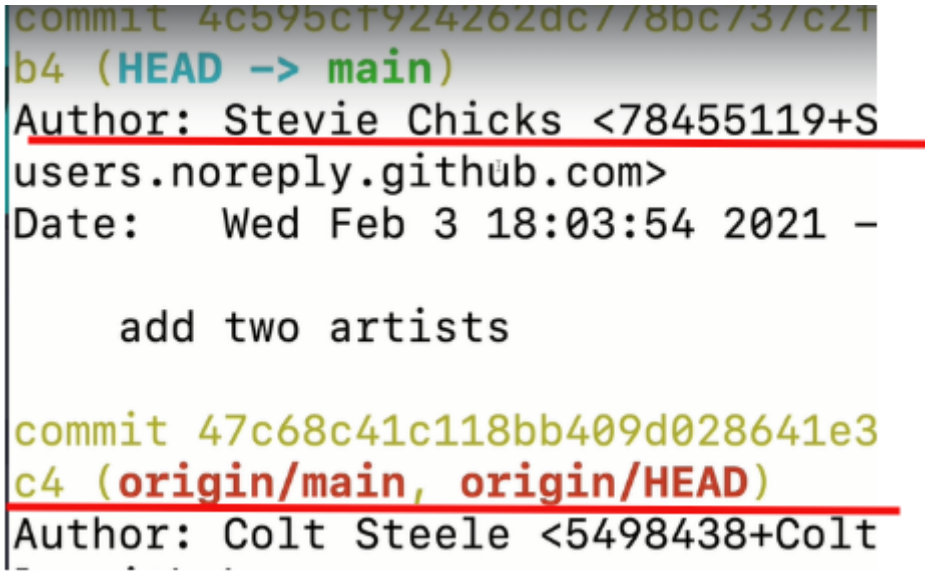




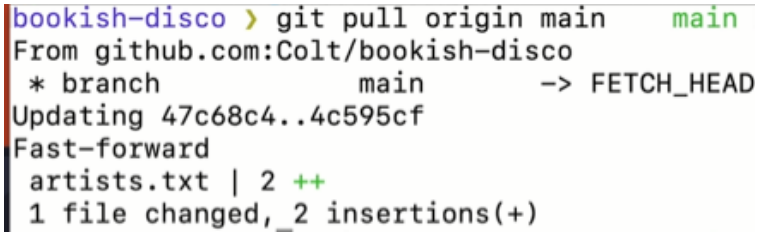
Adding Github Collaborators



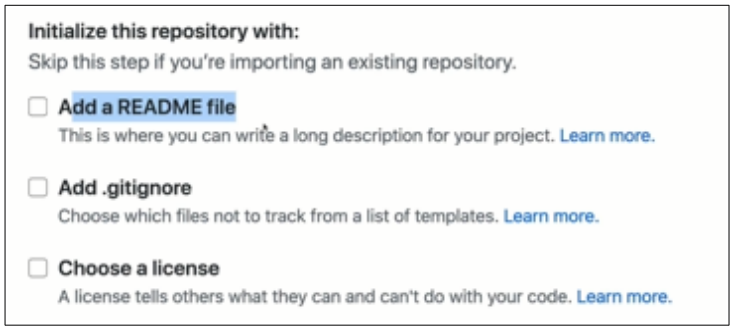
Github Collaboration Demo

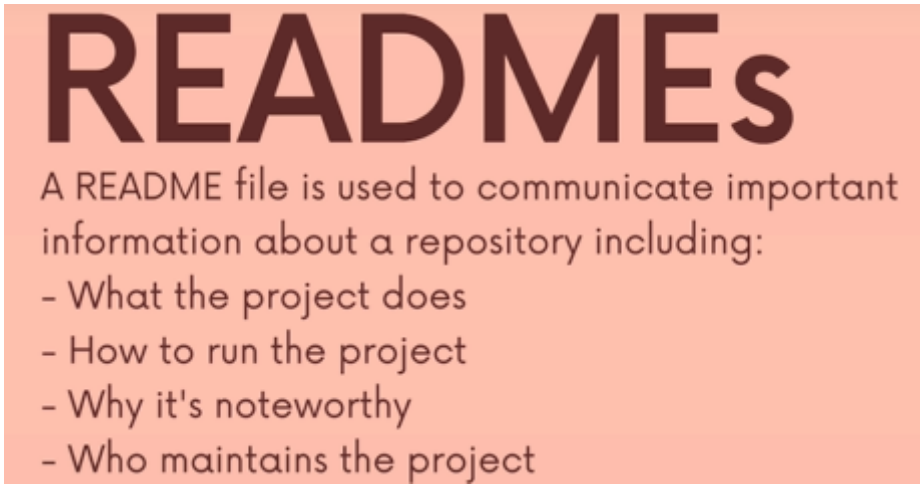


stevie will push the changes to colt github. Since stevie is a collaborator.



What are READMEs?



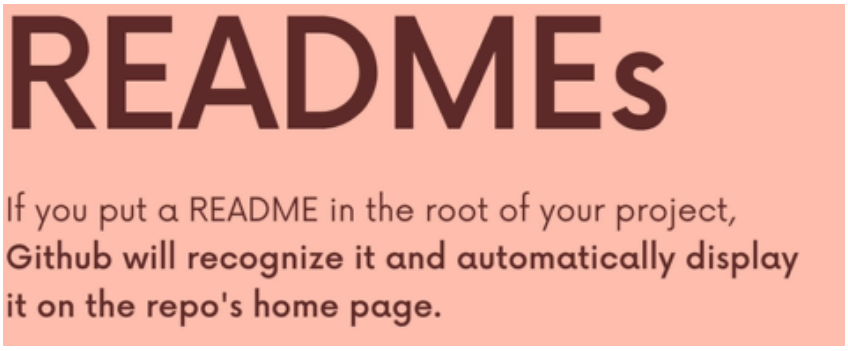


What your project does, whether you working on some API, or some Web Applications, video games. Even if you r just government using github to make law legistativley.

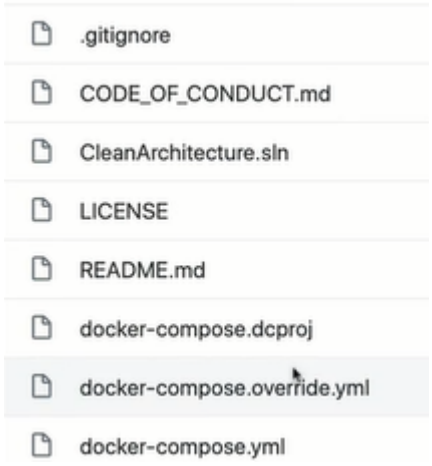
Whatever your repos is, the readme file should include information about what the project does, what its for, who is it for. How to get started with ur project. How to run and install dependencies if necessary. Why its noteworthy. Who maintains the project, how to collaborate the project, how to get help.

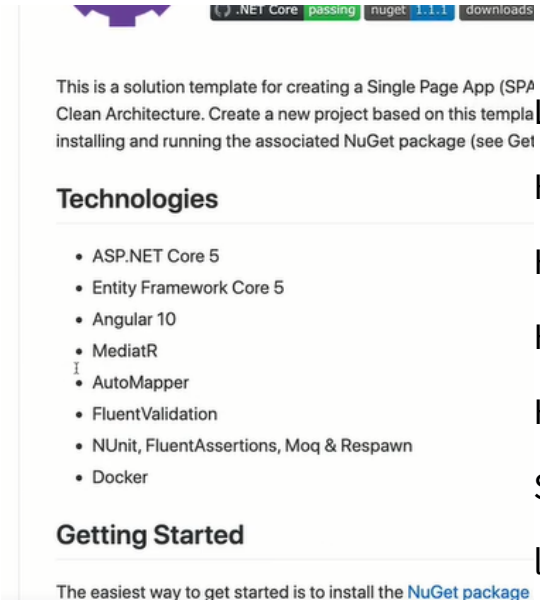
Not everything needs to be there. But if there is important info that you need to convey.

Somebody comment to your project for the first time, then put it in the readme file.



It will render





List of technologies.

How to get started

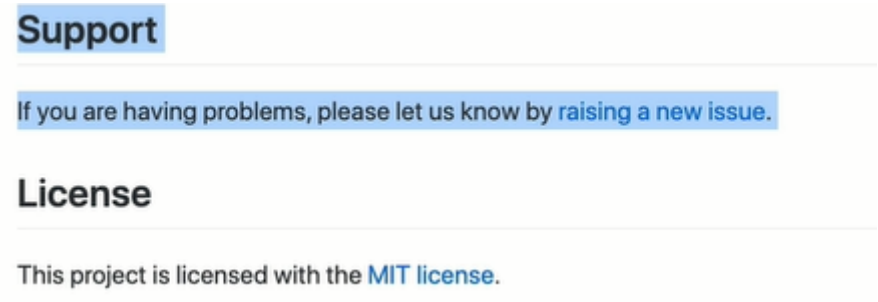
How to configure

How to configure db

How to migrate.

Support

license



Link for above: click

Link: Click

spaCy models

This repository contains releases of models for the spaCy NLP library. For more info on how to download, install and use the models, see the models documentation.

> ⚠ **Important note:** Because the models can be very large and consist mostly of binary data, we can't simply provide them as files in a GitHub repository. Instead, we've opted for adding them to releases as `.whl` and `.tar.gz` files. This allows us to still maintain a public release history.

Quickstart

To install a specific model, run the following command with the model name (for example `en_core_web_sm`):

```
python -m spacy download [model]
```

- spaCy v3.x models directory

Readme file always place at root of your project always. Sometimes .git folder.

Link: Click

React · license MIT npm v17.0.2 circleci passing PRs welcome

React is a JavaScript library for building user interfaces.

- **Declarative:** React makes it painless to create interactive UIs. Design simpl application, and React will efficiently update and render just the right com Declarative views make your code more predictable, simpler to understan
- **Component-Based:** Build encapsulated components that manage their ow complex UIs. Since component logic is written in JavaScript instead of tem through your app and keep state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest develop new features in React without rewriting existing code. React can a and power mobile apps using React Native.

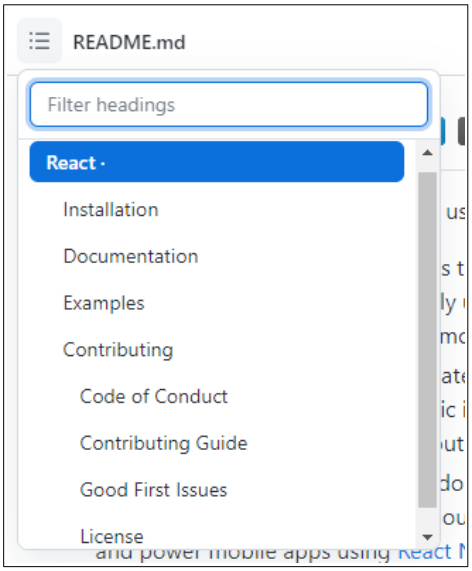
README.md

READMEs are Markdown files, ending with the .md extension. Markdown is a convenient syntax to generate formatted text. It's easy to pick up!

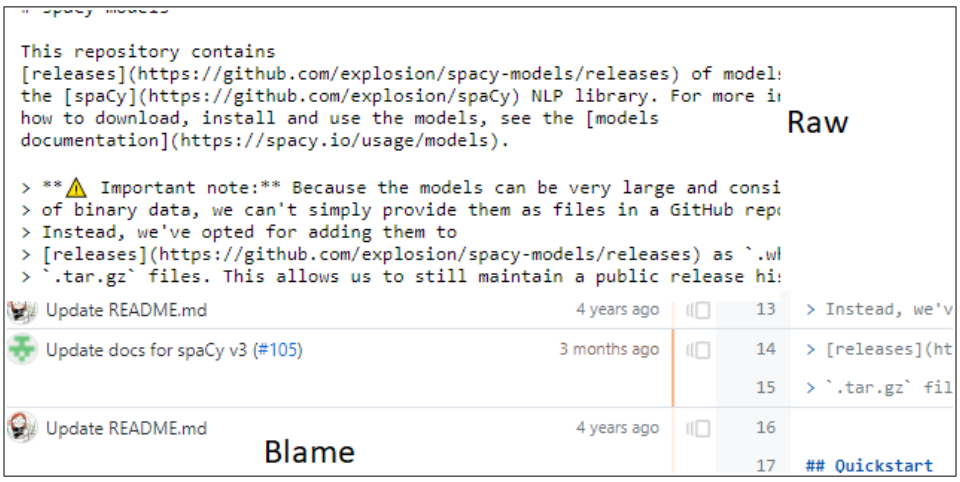
A Markdown Crash Course

Link:[Markdown webpage]

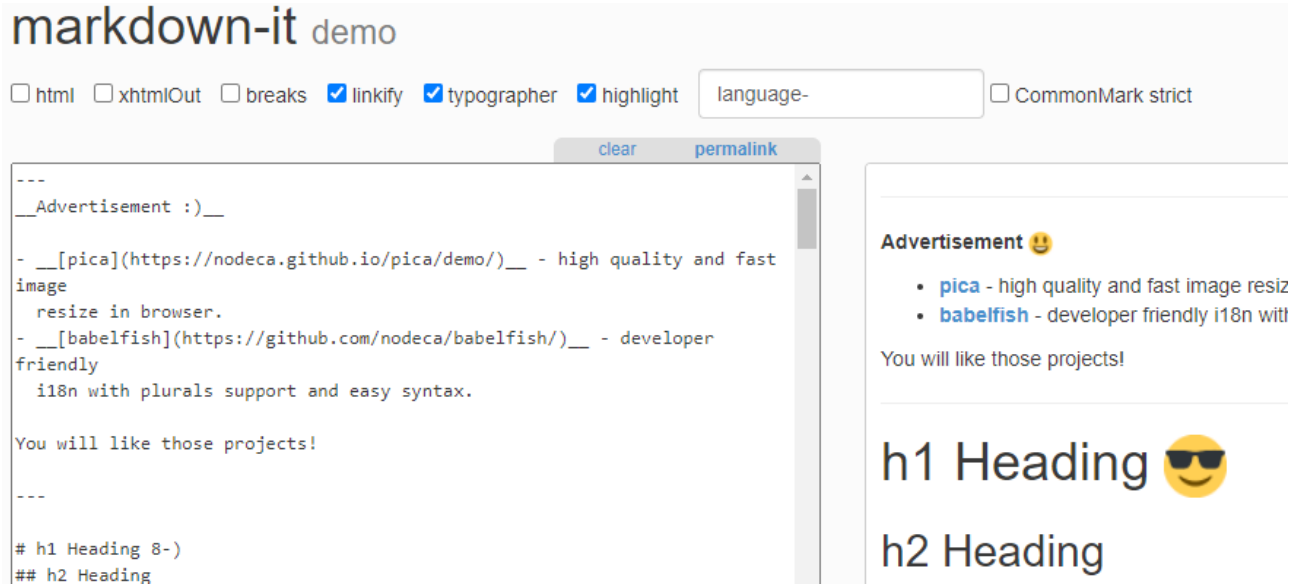
Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

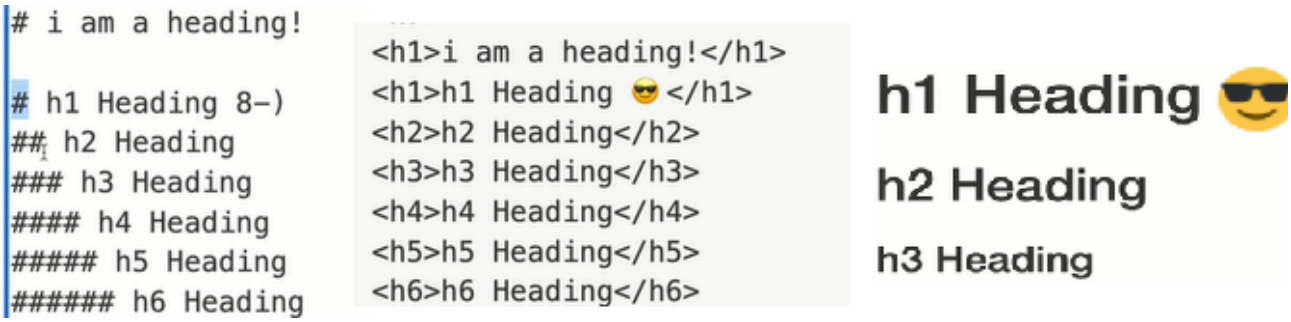


← how much structure is important.



Link:[Markdown-it](#)



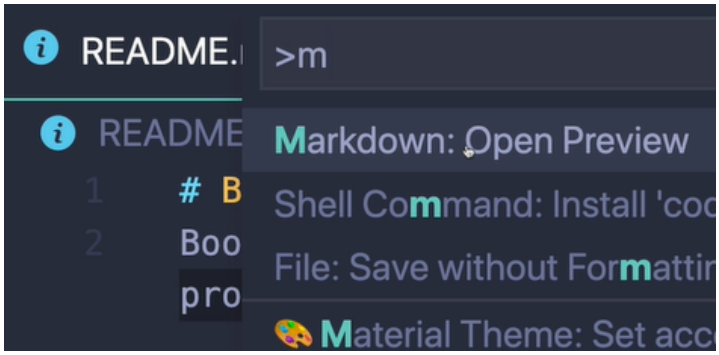


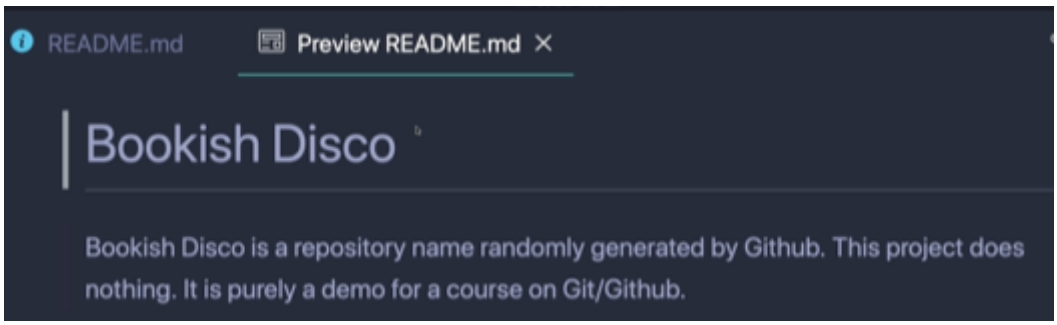
Reddit use markdown.

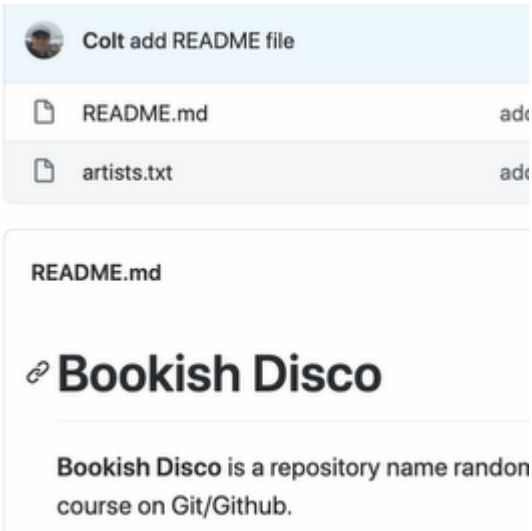
Adding a ReadMe To a project

Alot of time, you working on a project. You don't expect for anybody to ever see it.

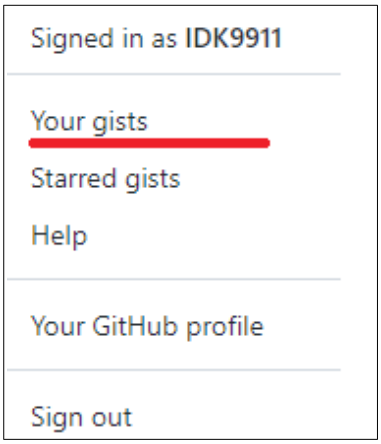
They are the probably the one thing, that a recruiter or a non technical person would see, if they are looking at your projects.

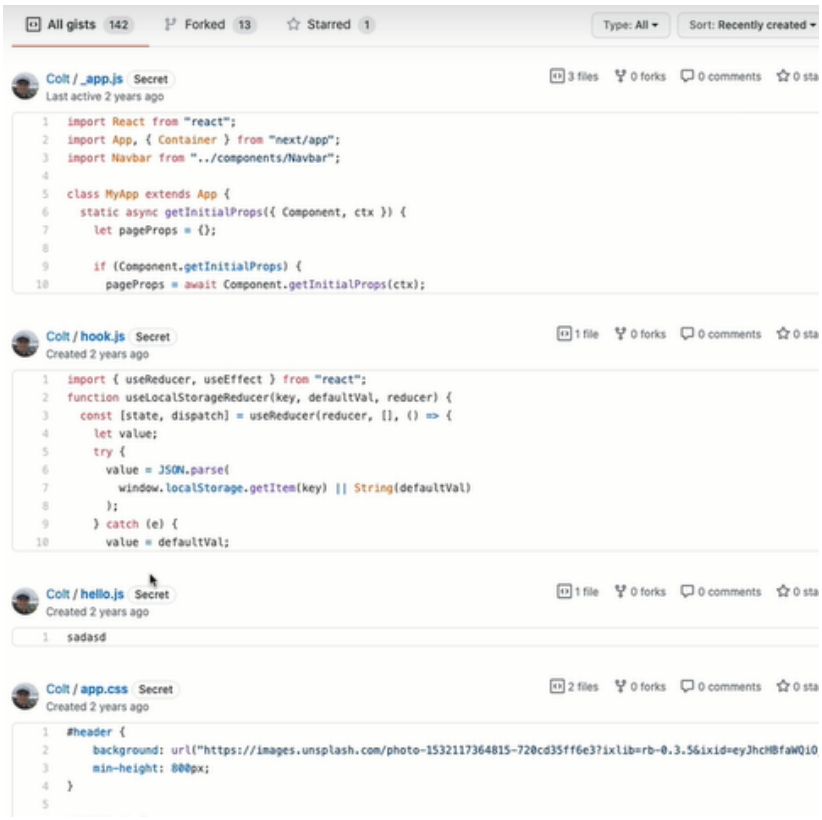






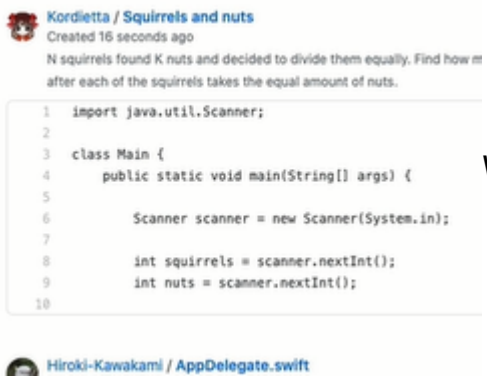
Creating Github Gists





The ability to leave comments, I can personally delete this file, edit this file. I can download it and share with people, fork it and star it

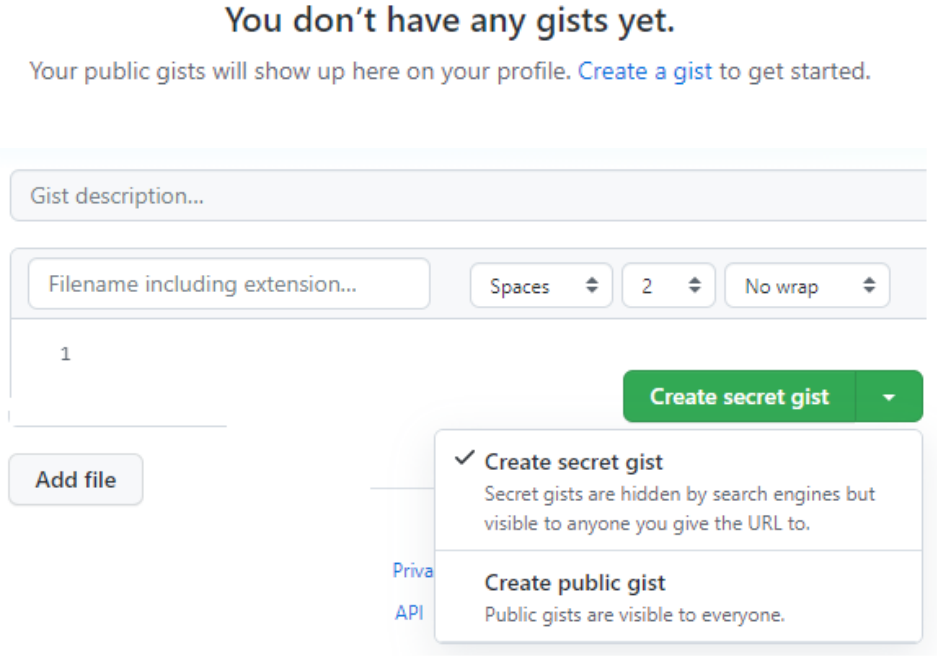
All gist



You can see gist what other people have made.

So they range from mark down files.

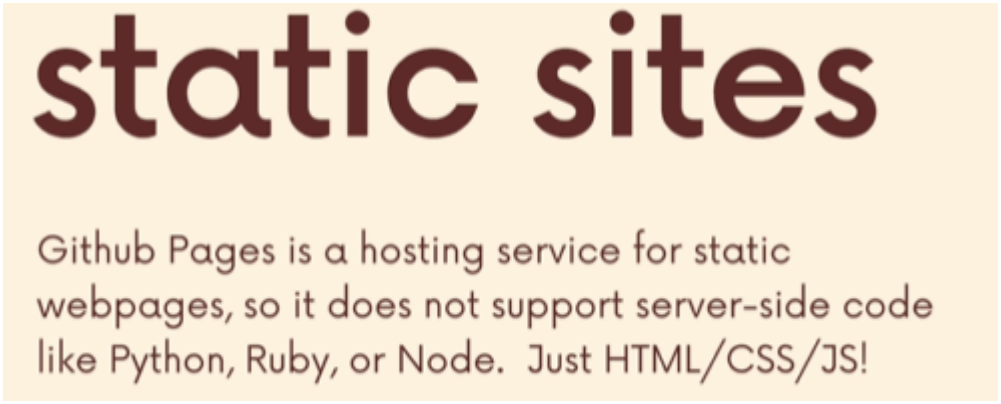
We use git for teaching and share stuff.

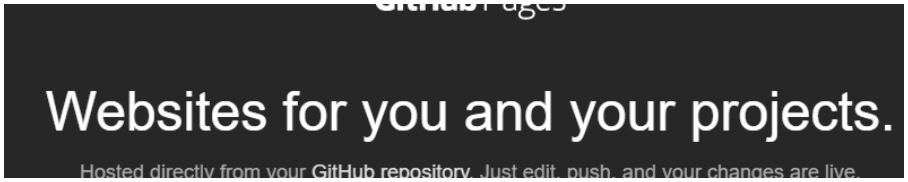


Introducing Github Pages

pages.github.com

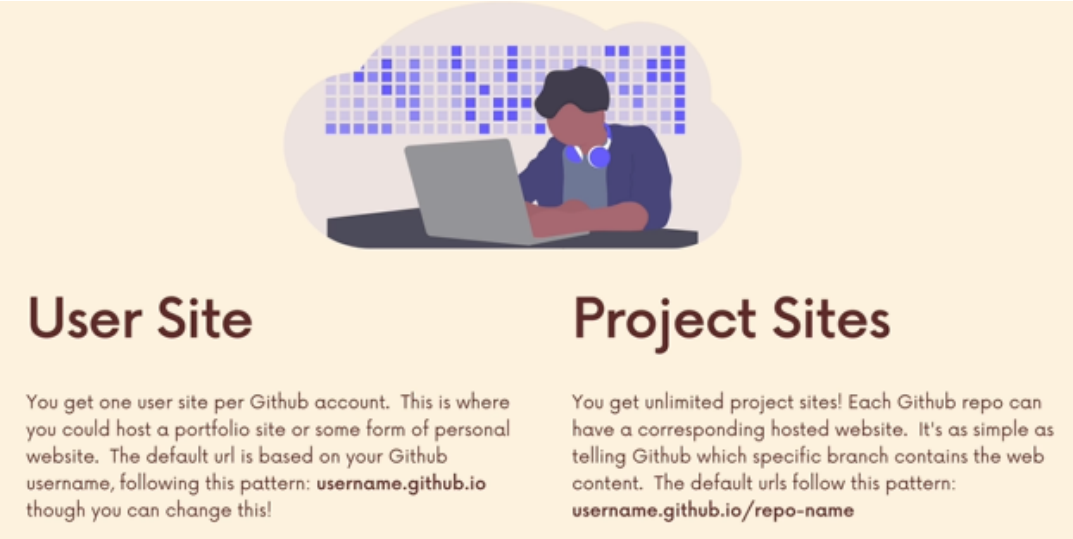
Link: [Click]



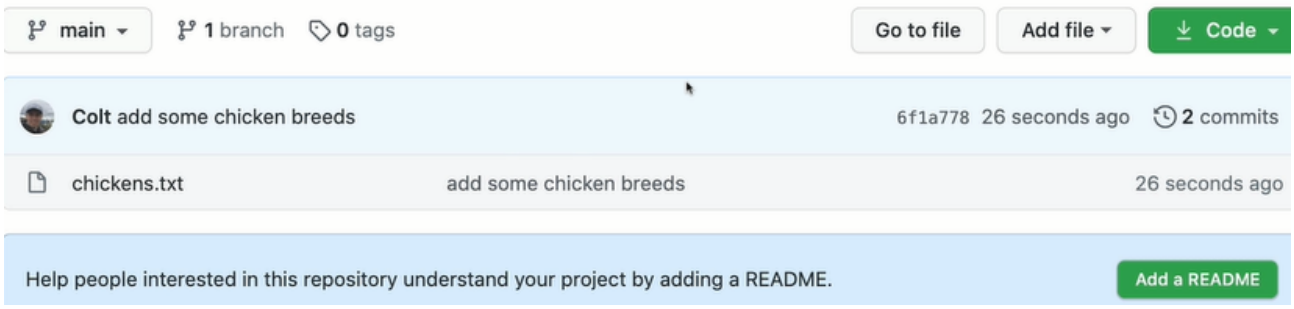


For static web sites, for portfolio websites or documentation website for your repositories. If you build a js game , you can use github pages to host that.

You don't expect github pages for high capacity web traffic. Its not really meant for that. Its free and its built for github.

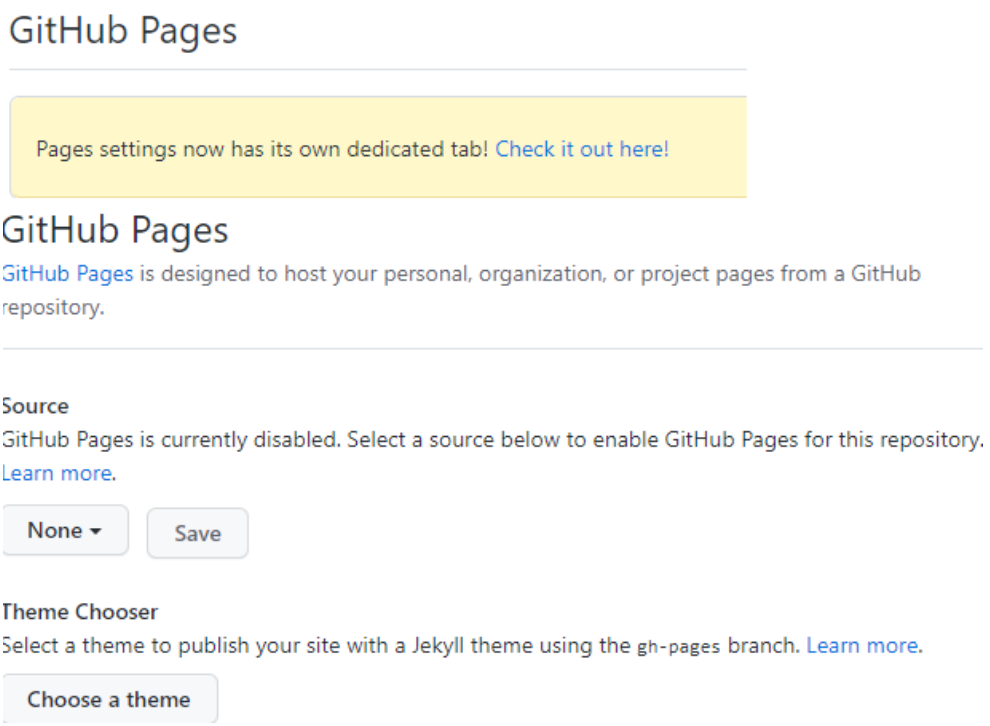


Github Pages Demo

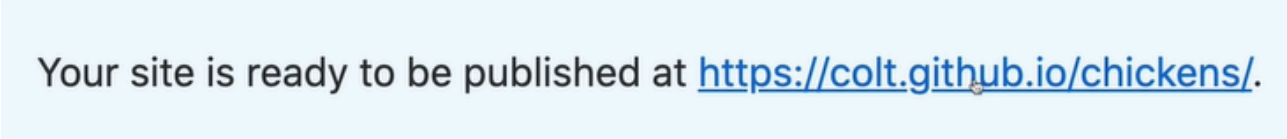


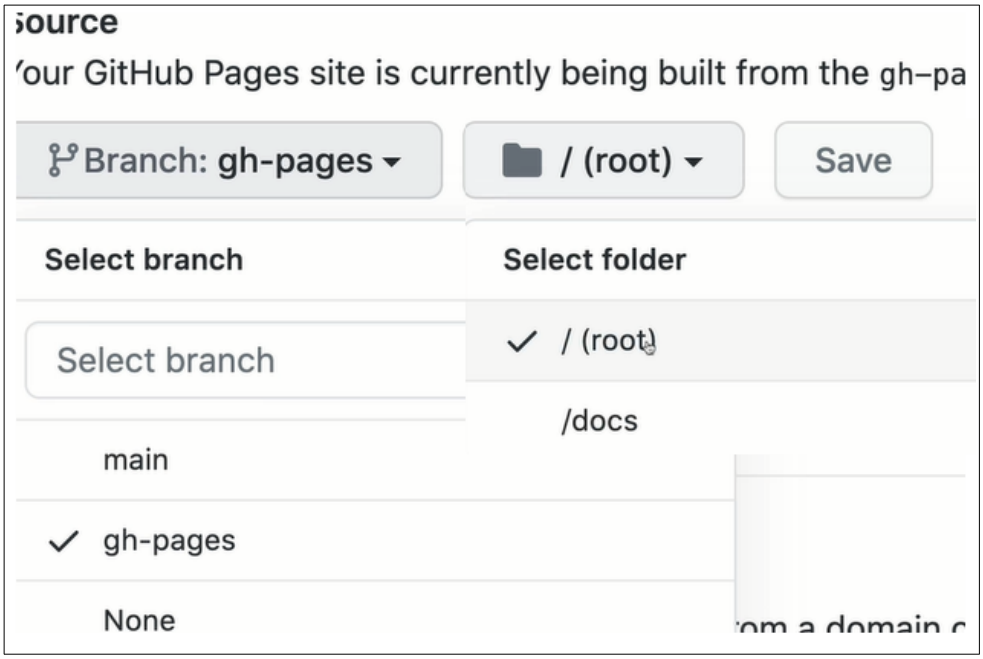
Now , if I have to setup github pages website, all that I have to do is select some branch, I am gonna tell github contains a website.

Its gonna look for a file name “index.html”



You can setup your website on any branch whether its main or not.



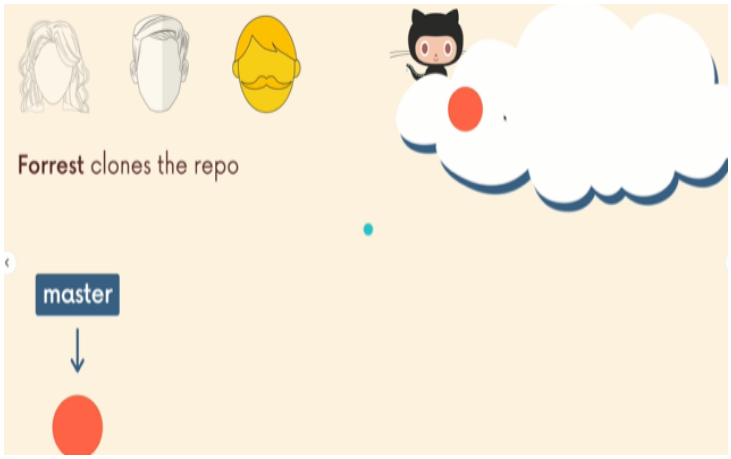


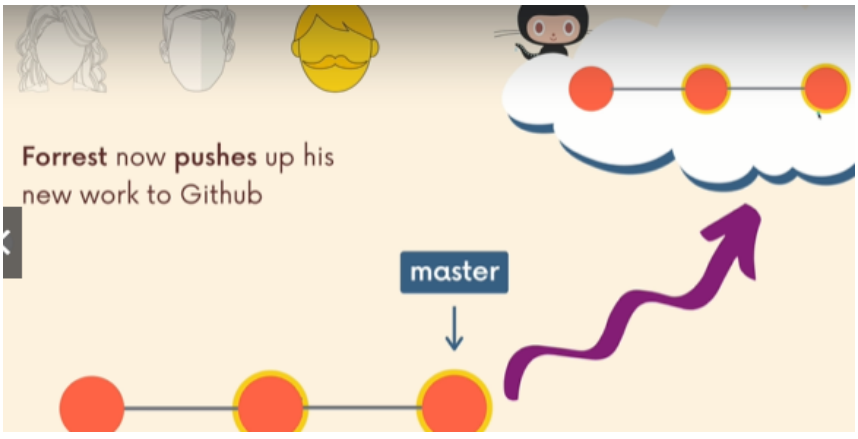
gh-pages are standard defualt names.

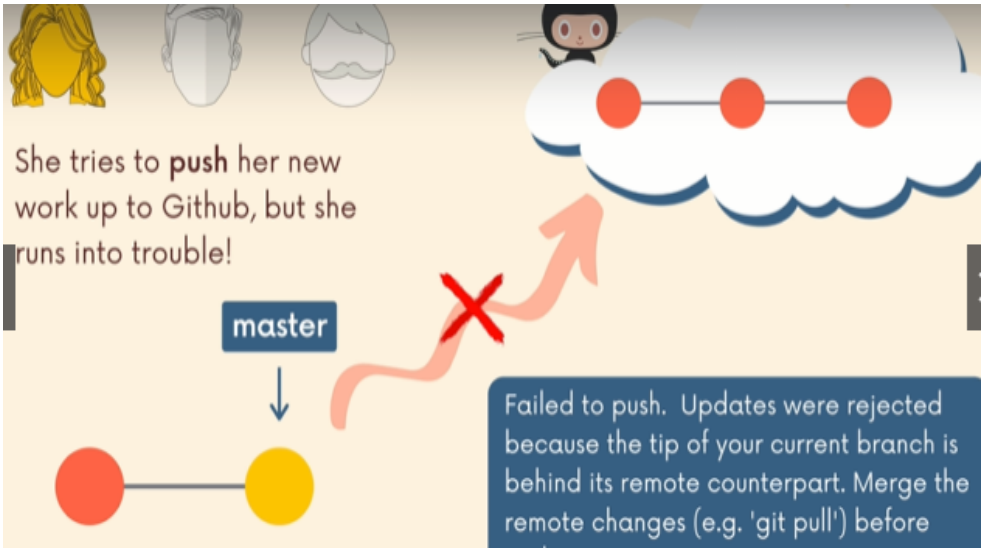
Git Collaboration Workflows

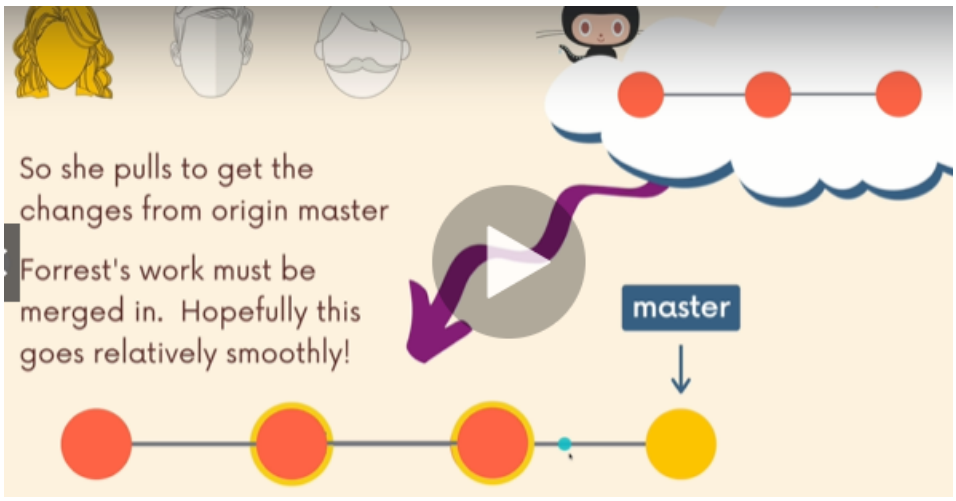
Different ways of configuring repositories, differents ways of working on branches, When to create branches, how to merge changes, how to ask to merge changes How to get permission.

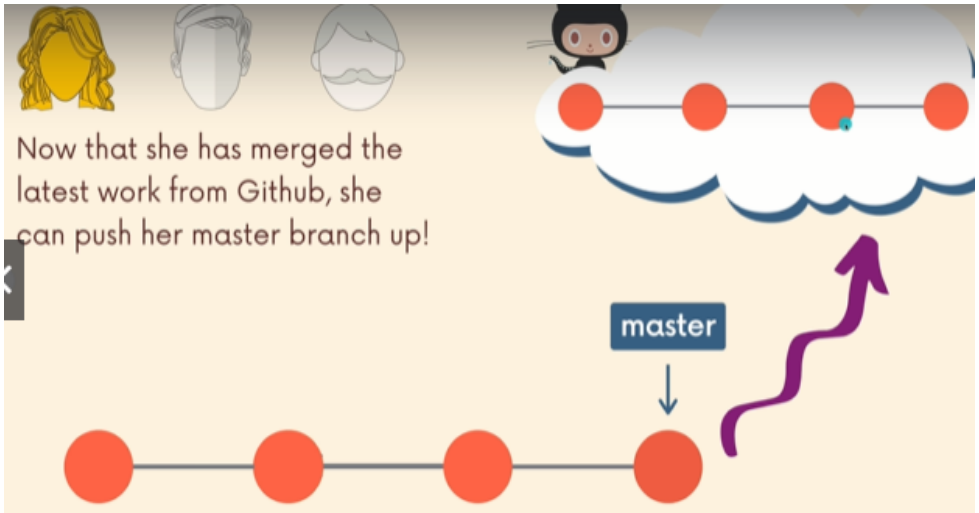
The Pitfalls of a Centralized Workflows

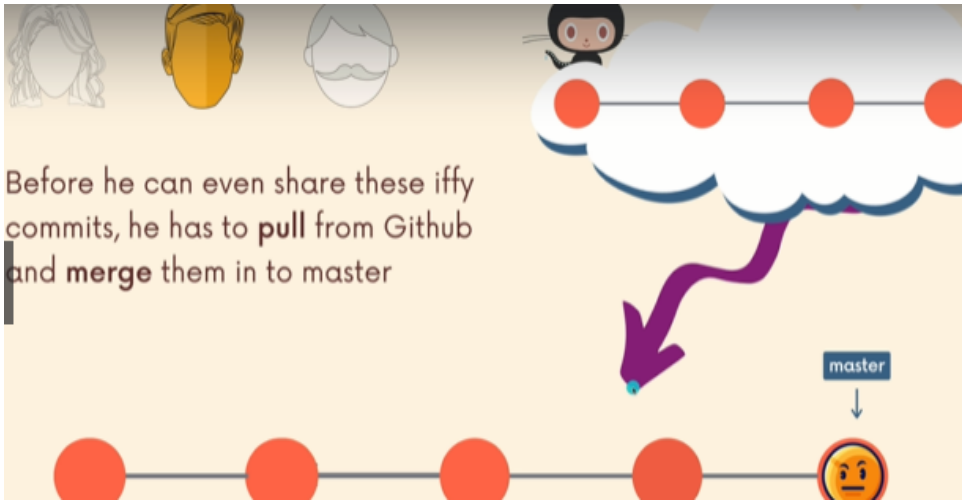




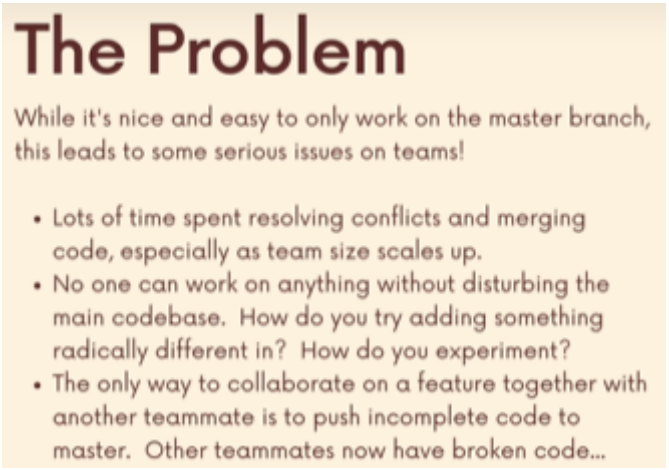

She tries to **push** her new work up to Github, but she runs into trouble!
master
Failed to push. Updates were rejected because the tip of your current branch is behind its remote counterpart. Merge the remote changes (e.g. 'git pull') before


So she pulls to get the changes from origin master
Forrest's work must be merged in. Hopefully this goes relatively smoothly!
master


Now that she has merged the latest work from Github, she can push her master branch up!
master



if this is incomplete, so everyone has a incomplete code. He broke the codebase.



```
centralized-workflow ❯ git push origin main
To github-stevie:Colt/centralized-workflow.gi
t
 ! [rejected]        main -> main (fetch firs
t)
error: failed to push some refs to 'git@githu
b-stevie:Colt/centralized-workflow.git'
hint: Updates were rejected because the remot
e contains work that you do
hint: not have locally. This is usually cause
d by another repository pushing
hint: to the same ref. You may want to first
integrate the remote changes
hint: (e.g., 'git pull ...') before pushing a
gain.
hint: See the 'Note about fast-forwards' in '
git push --help' for details.
```

Above is Central workflow problem. Stevie is tryna push is broken code in the master branch while his fellow coworker has already made commits on the remote repo. So stevie is behind some “n” commits. In order to show his coworkers and start discussion he has to merge “source of truth” or “official branch”.

Leading to “broken code” at “source of truth”

```
centralized-workflow ❯ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done
```

```
centralized-workflow ❯ git status          main
On branch main
Your branch is ahead of 'origin/main' by 3 co
mmits.
  (use "git push" to publish your local commi
ts)
```

```
centralized-workflow ❯ git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 2.14 KiB | 2.14
```

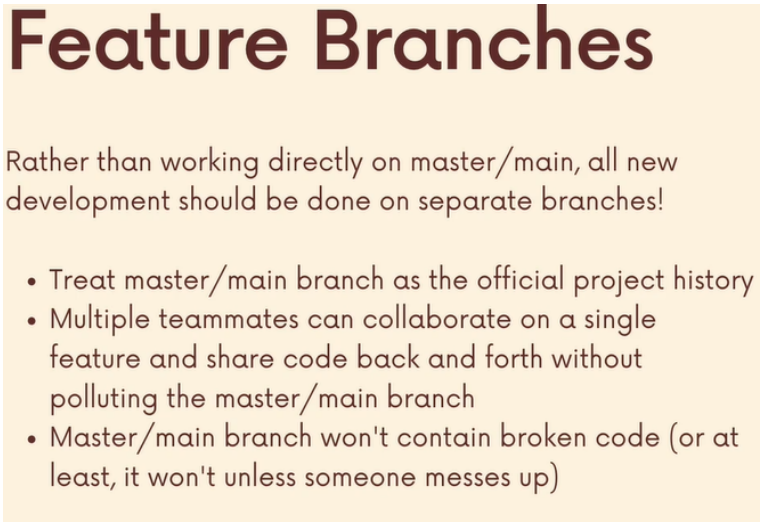
Next day, colt receive changes from stevie to help with his code.

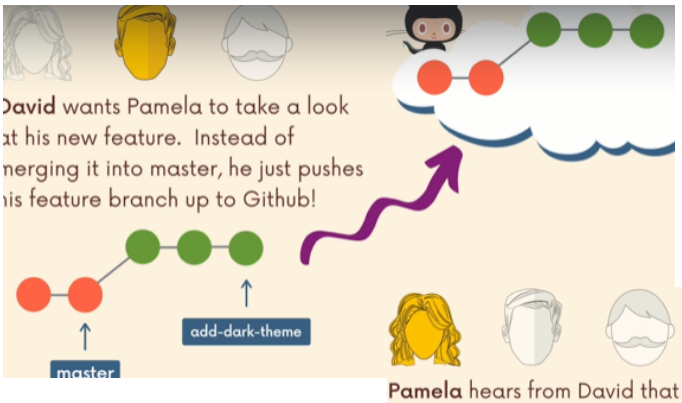
He has changes in his local repositories which arent commited.

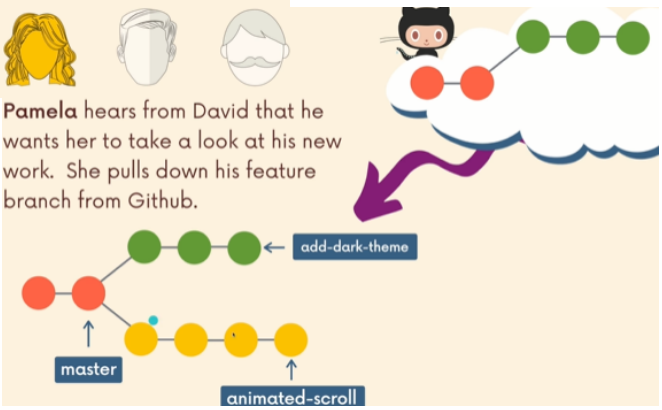
```
        modified:   index.html

no changes added to commit (use "git add" and
/or "git commit -a")
```

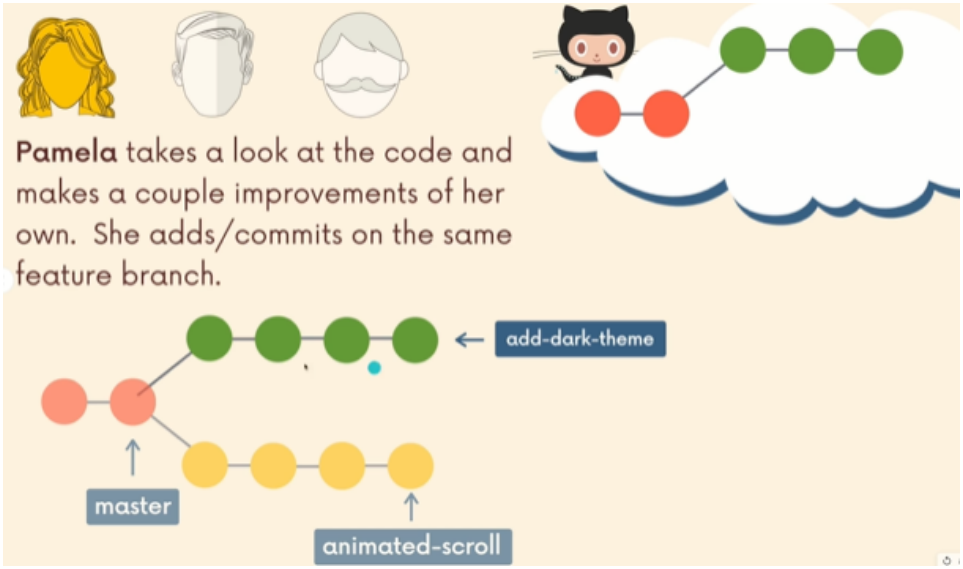

He needs to stash those changes, since he don’t wanna commit yet.

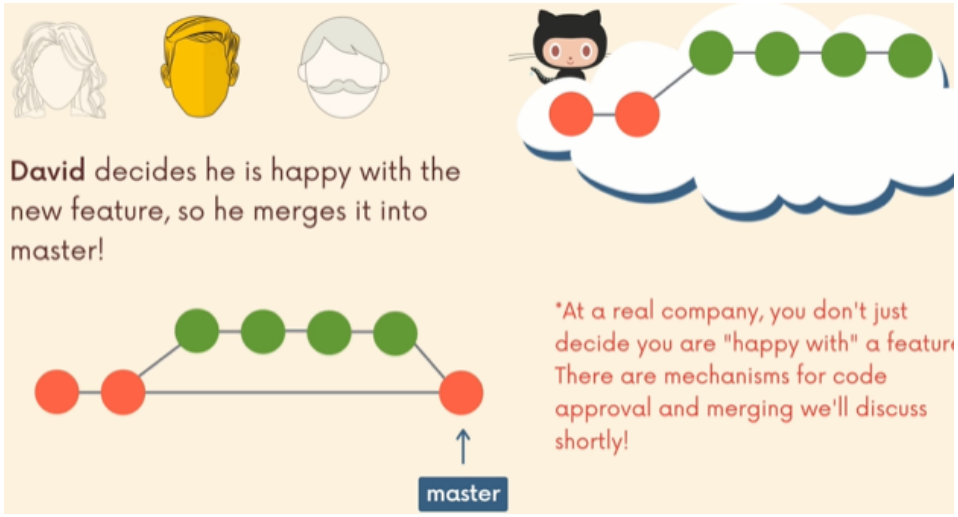
The All-important Feature Branch Workflow

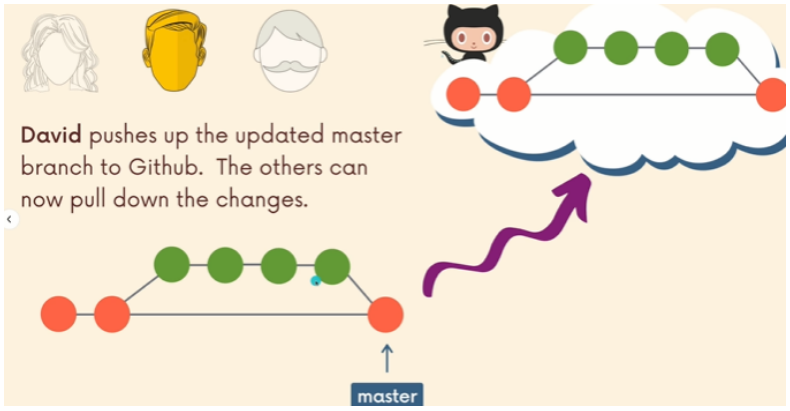












we dont merge broken stuff in the main branch.
Unlike the centralized workflow, so everyone works on a branch.
You can heva multiple branches too. It's not necessarily like Pamela's branch and David's branch. It's more one branch per feature or bugfix.
Branches are feature oriented.
we share those branches, collaborate on branches.
And keep our master or main branch as our source of truth. As a stable, always working and functioning , non-broken branch in our project.

these are pattern that org uses. its just a name, it could be called just navbar

Feature Branch Work Flow

We havent really talk about naming feature branch in general. For e.g

feat/navbar <!-- By feat he meant features

refactor/navbar

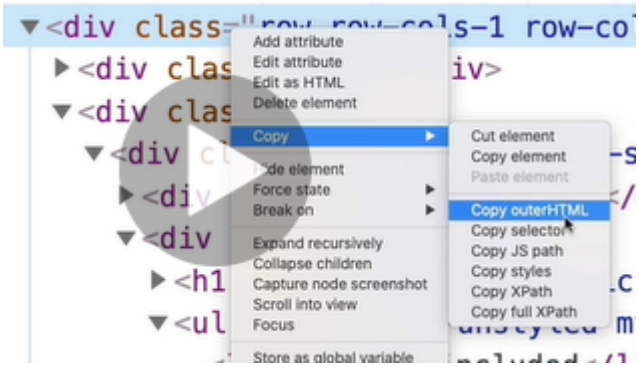
bug/navbar

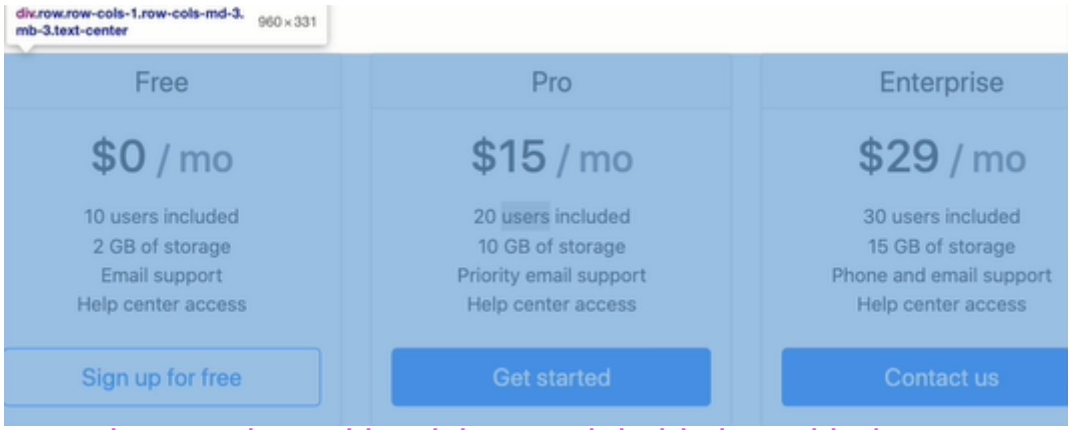
fix/navbar

```
feature-branch-workflow ❯ git pull origin mai
n
From github.com:Colt/feature-branch-workflow
 * branch            main       -> FETCH_HEAD
Already up to date.
```

Its always a good idea, to see is there any new commit. If there is, you can start your day with fast forward.

```
feature-branch-workflow ❯ git switch -c prici
ng-table
Switched to a new branch 'pricing-table'
```





its a project with pricing model, this isnt github

Advantage of feature branch is that in contrast to centralized workflow, we wont have to merge code with others on the main branch(sot) just so we can collaborate who are working with us to fix the bug.

Everytime you make a feature branch, you don't have to push into github.

Alot of time , you wont do that. Unless you have a particular reason to share it to somebody.

If you look at organization repositories, they are not gonna be 1000s of feature branch. Even though there have been 1000s of feature branches over a couples of months.

Alot of time, we remove then.

Alot of time, we merge it to master or main sometime or delete the branch when you done.

```
feature-branch-workflow ❯ git fetch origin
remote: Enumerating objects: 5, done.
```

To see if ,Has anything changed.

```
Unpacking objects: 100% (3/3), done.
From github.com:Colt/feature-branch-workflow
 * [new branch]      navbar     -> origin/nav
bar
```

```
h Workflow Demo
origin/HEAD -> origin/main
origin/main
origin/navbar
origin/pricing-table
```

```
feature-branch-workflow ❯ git checkout origin/navbar
```

```
HEAD is now at fba59a8 add broken navbar
feature-branch-workflow ❯ git switch -          HEA
Previous HEAD position was fba59a8 add broken navbar
Switched to branch 'pricing-table'
```

i need to leave detached head.

```
feature-branch-workflow ❯ git branch    prici
feature-branch-workflow ❯ git switch navbar
```

```
feature-branch-workflow ❯ git commit -m "fix navbar bu
g"
```

```
feature-branch-workflow ❯ git push origin navbar
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
```

we have alot of options how we intend to do this, one is merge at will. whenever anybody wants to merge their changes in, just go for it.

Merging Feature Branches

Delete the branch after merge if we dont need it anymore.

Introducing pull request

So if you are in a company where the stakes are high, where you cant expect to screw up something in the main or master branch.

What if somebody did a bad commit or something that break the main branch. At the very least, you've got 100 more developes who now have a bad version of that main branch and they have to get rid of commit or reset it.

Its just annoying.

You want to make sure that whatever you are merging in or whoever is doing the merging has some discussion around it, has some approval.

Maybe changes are made at the request of somebody else.

There's some code review process involved and pull request are integral part of the process.

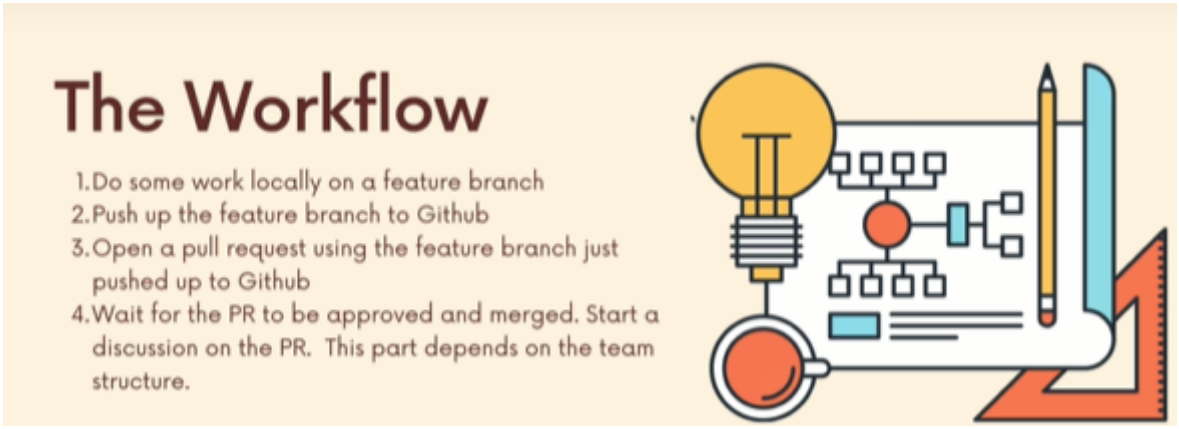
especially in larger teams with lots of collaborators and features. If there is no testing going on beforehand, if there is no discussion or converation before hand. that could get way out of control. And you would never do that on a real product with users and things at stake.

so another way is to merge your own branches in, but only after you've discussed. You've sent an email or chat message or some sort of conversation with your team and someway that whether you should merge those changes in or not.

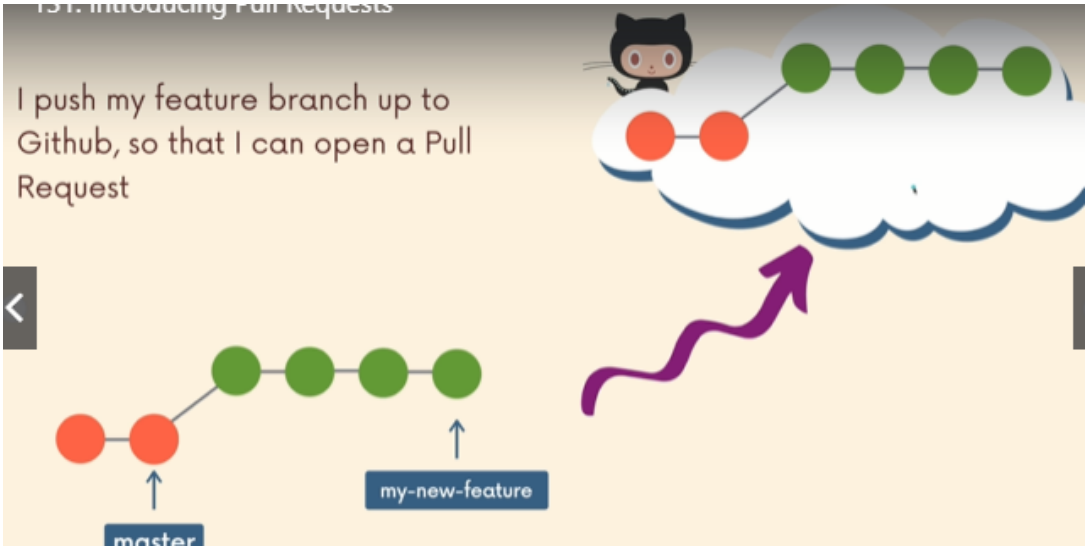
3rd option is pull request:
we've talked about merging feature branches to the defualt branch/(sot)/main and how much it has made our life easier. But then at some point somebody has to merge those changes in from the feature branch into main branch.
so rather its better to have some code review process before someone merges their bad commit. so to protect the branch and keep it functioning, working at all times.
we have PR.
So u have to make sure, whatever u are merging in, or whoever is doing the merging have some discussion around, it has some approval.
Maybe changes are made at the request of somebody else. Basically taking in feedback.
So PR are not a native git feature. They are built on top of platforms like Github and bitbucket and some others.
Not exclusive to github and not 100%part of the git itself.

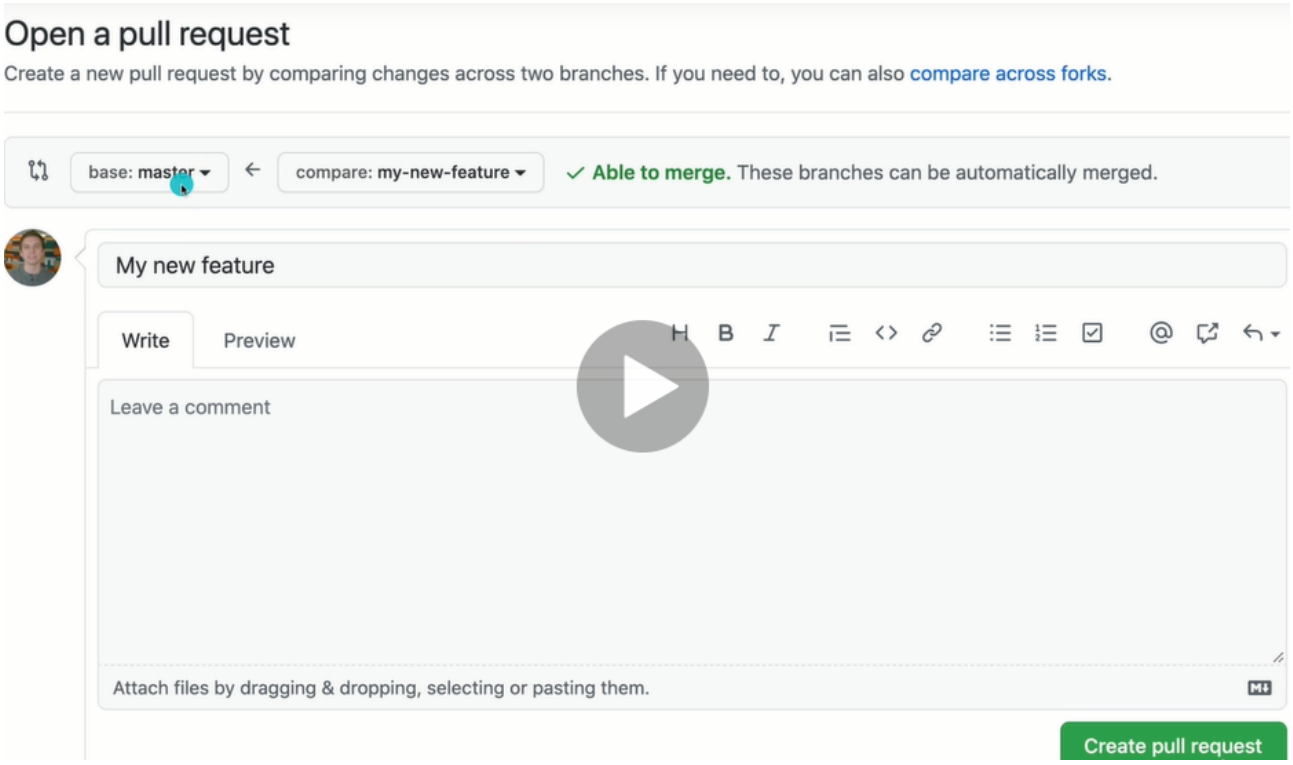
PR:
inform other collaborators that they have new work that needs to be reviewed before integrated.
so we have discussion,feedback,update the work on the same PR.

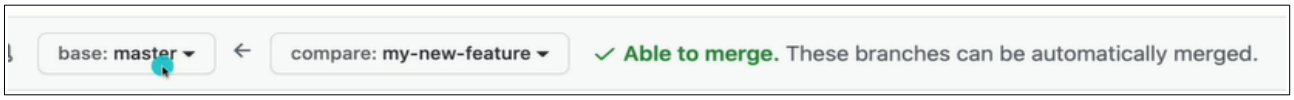
So, Hey, everybody on my team, i have this new branch with this feature i completed or i think its complete and its on github. Im making this PR, can u take a look and somebody merge it in if its good or tell me if its not good.
The main branch is protected usually, u have to make a pull request.



Maybe multiple ppl have to approve.

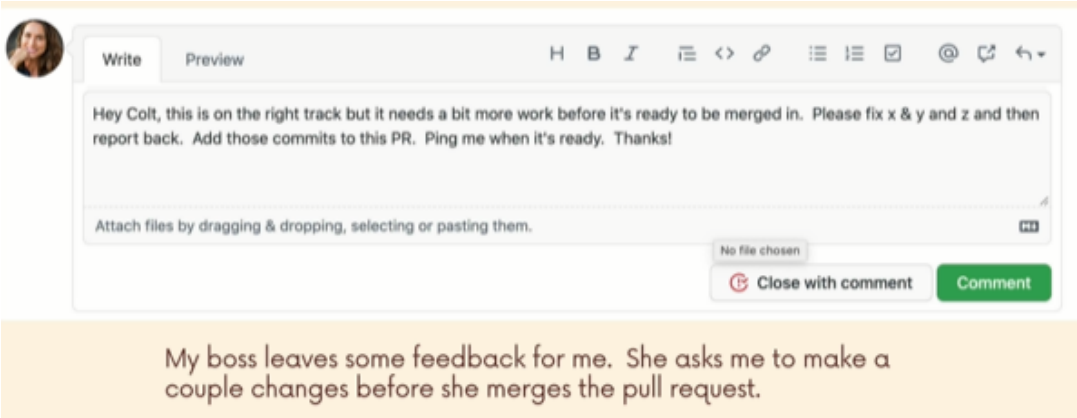






I am requesting my-new-feature to merge into master.

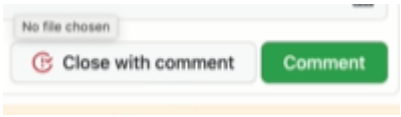
Then i need to provide some information, i need to come up with what my pull request is. Whats the reason for it. What does it includes. Depending on the company and organization. Theres gonna be pretty strict rules for what goes in here.



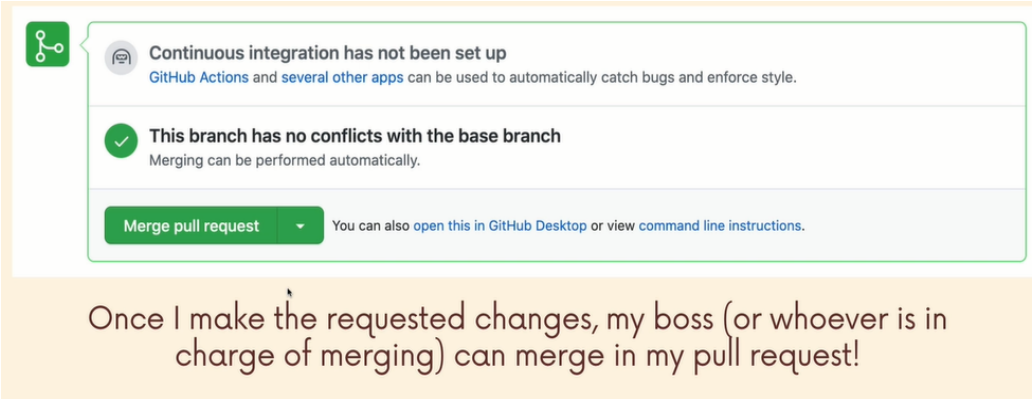
I as an employee dont have the ability to merge directly into master in github. I am supposed to make a pull request.

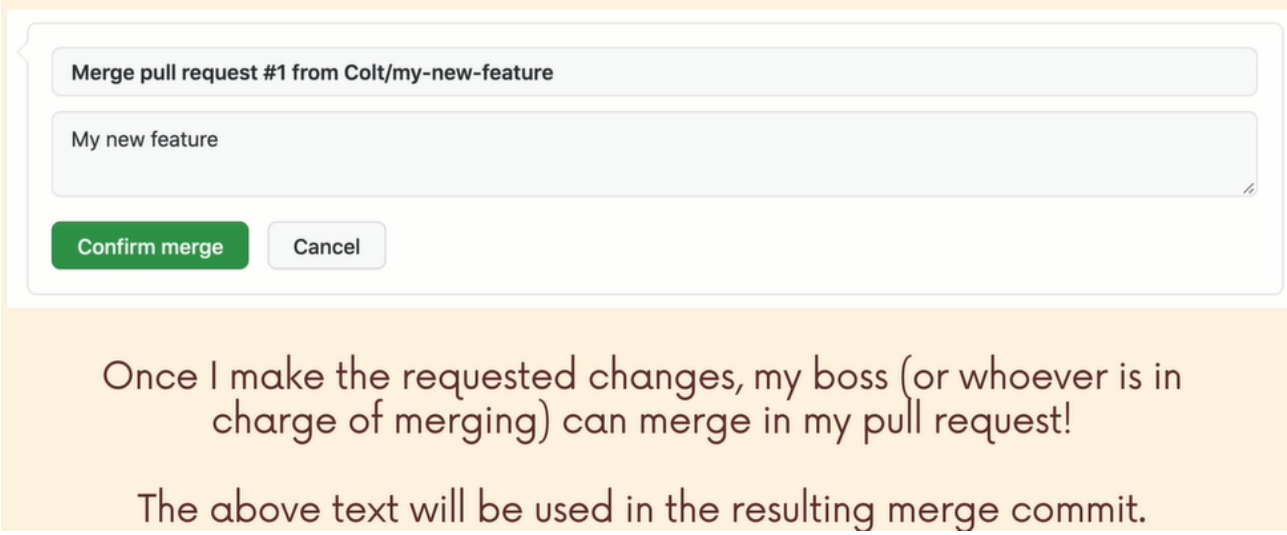
You can ask help of others in ur teams who are reviewing the PR. U can mention them in the comments.

My boss find some issues, leaves me some feedback. She asked me to make some changes, ping me when its ready. I can notfiy ppl and mention ppl in comments and i can attach files and she can also just closed it, its kinda blunt way of saying no.



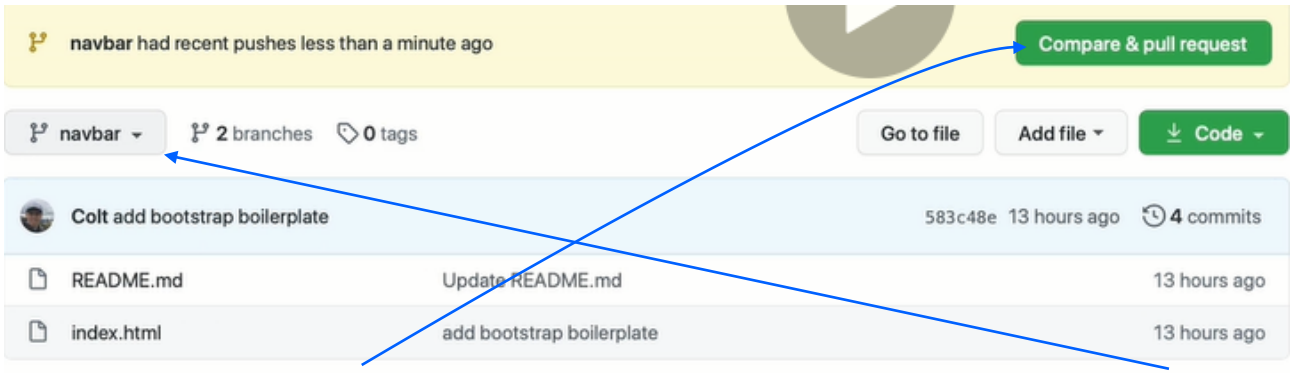
Colt: hm, r u sure you want me to do that. Whats the motivation and im not sure that i agree with that. What do u think, other developers. I can bring somebody else in and can have a full discussion on the team.





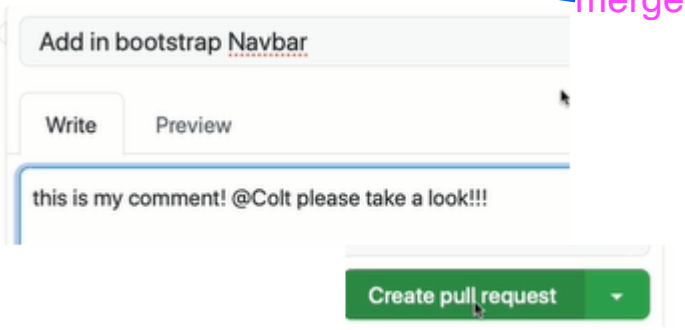
Note: u dont have to create a new PR. if they give u some feedback or tell u to make some changes.

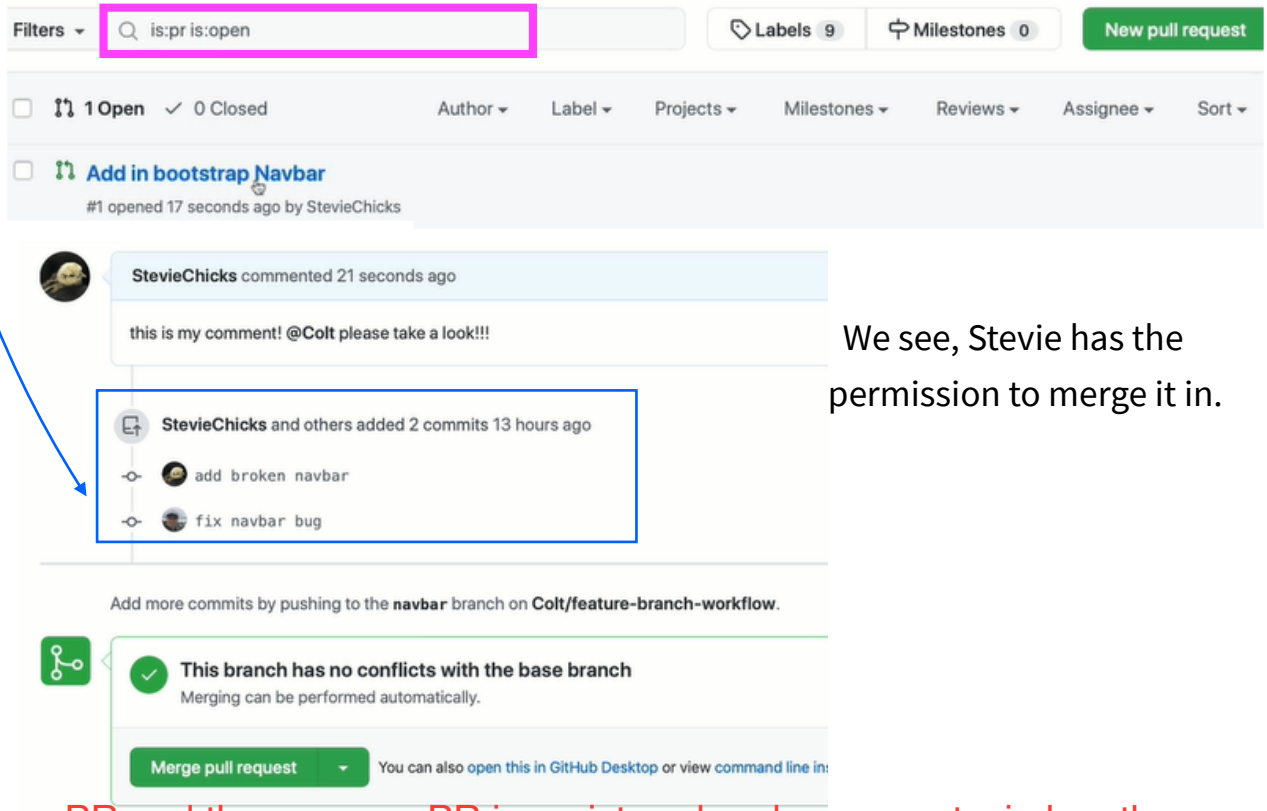
Making our First Pull request



To make a PR request click compare & PR on ur feature branch

merge feature branch on main

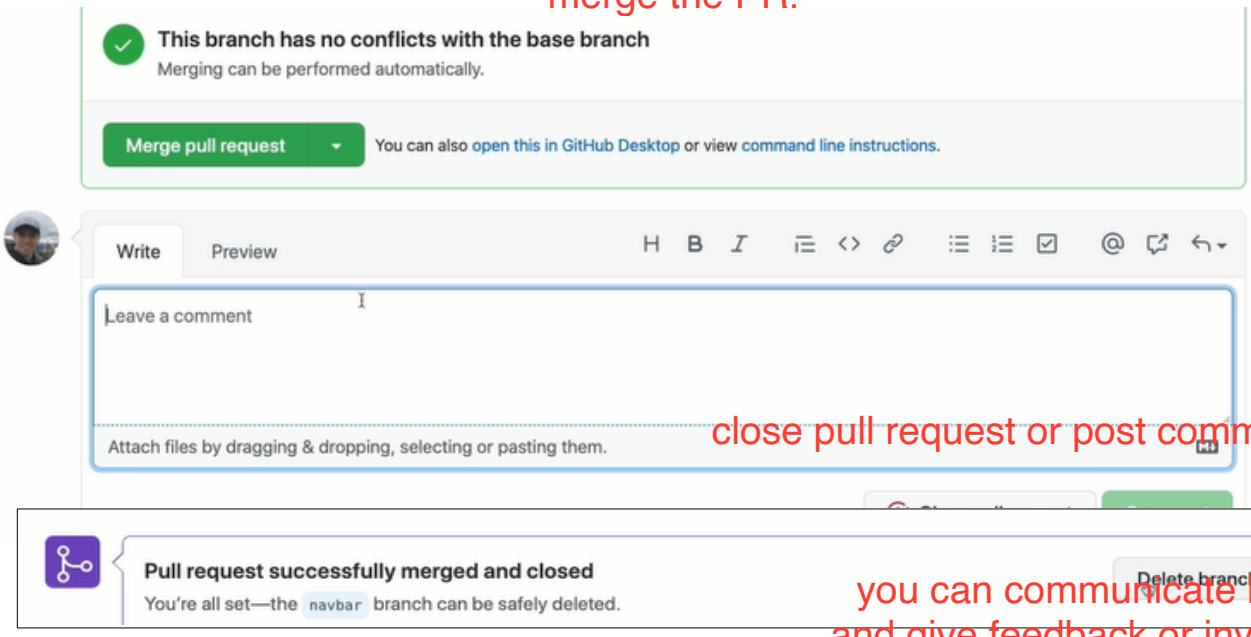




We see, Stevie has the permission to merge it in.

we make a PR and then we see PR is registered and we saw stevie has the permission to merge in the PR. This is not a recommended way. Merge approval or a team is required.

here, colt is from the pr approval team. we see colt image(not stevie the pr maker). since colt was mention in the PR he also get notified. he can close or merge the PR.

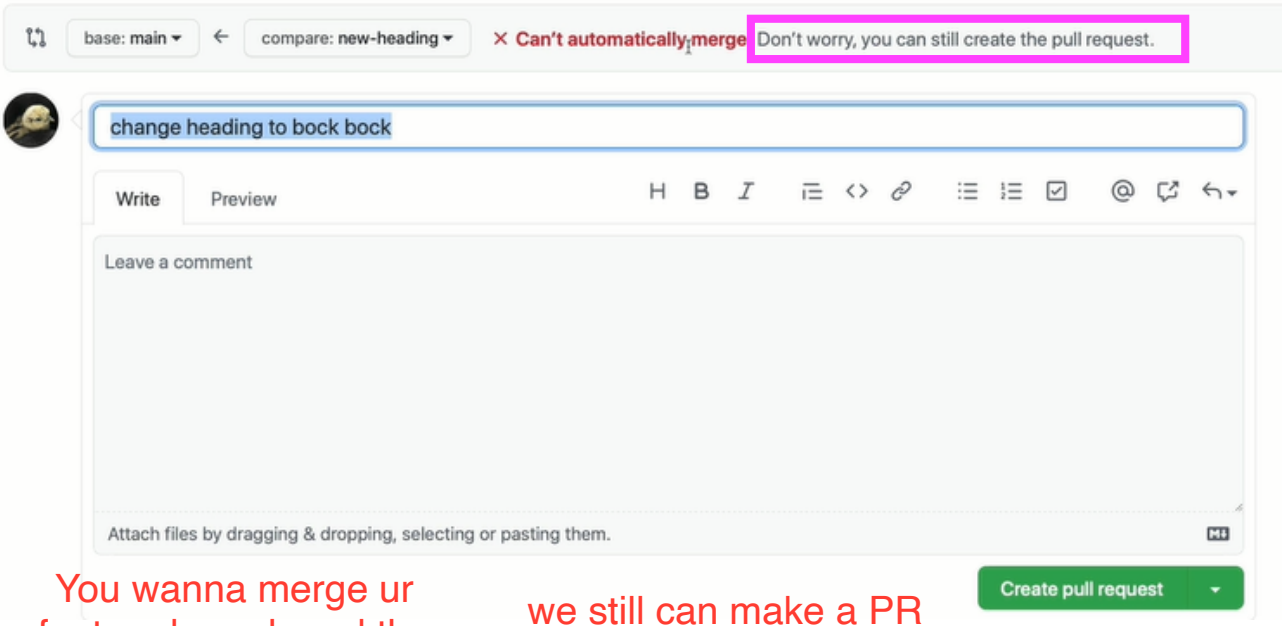


close pull request or post comment

you can communicate here first and give feedback or involve more people or say looks good merging in 10mins.
All this before merging PR.
So u basically now goto ur local machine and pull the changes. Then if u like u can delete branch from github and then the git.

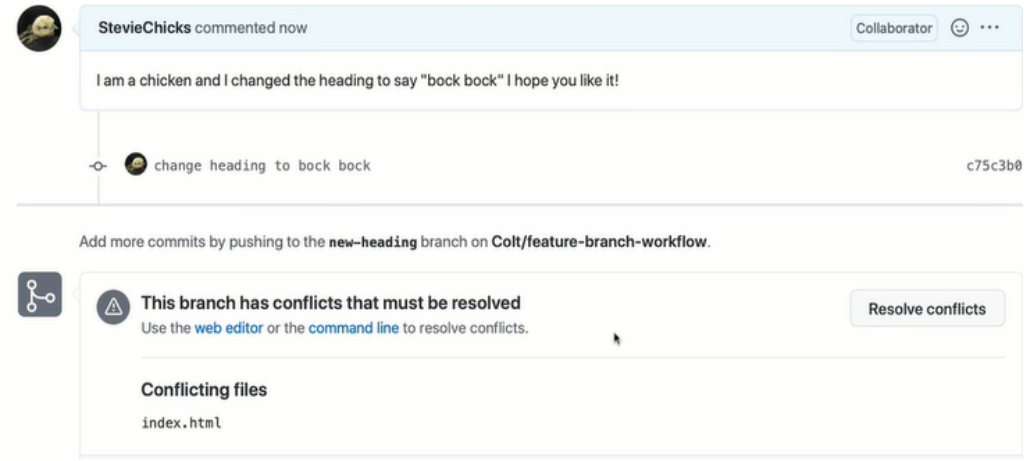
```
feature-branch-workflow ❯ git switch
main
Switched to branch 'main'
Your branch is up to date with 'origi
n/main'.
feature-branch-workflow ❯ git branch
-D navbar
Deleted branch navbar (was 47bd0b5).
```

Merging pull Request with Conflicts

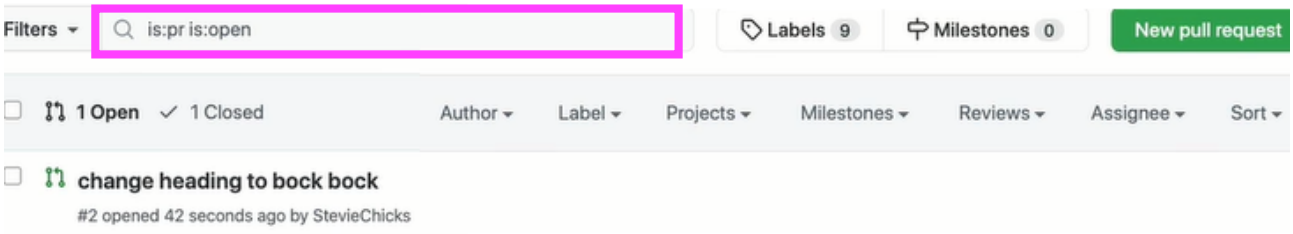


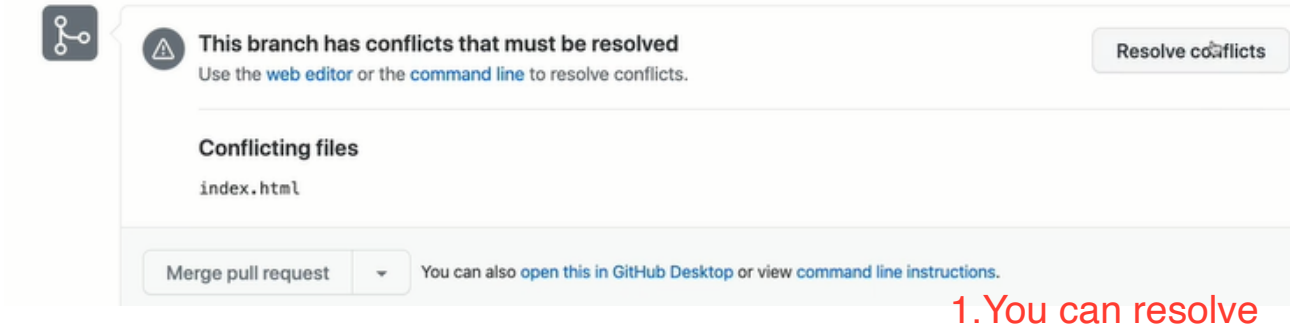
You wanna merge ur feature branch and the main branch. But there's a conflict

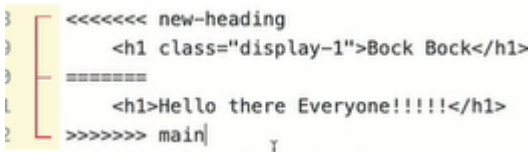
we still can make a PR



Two ways

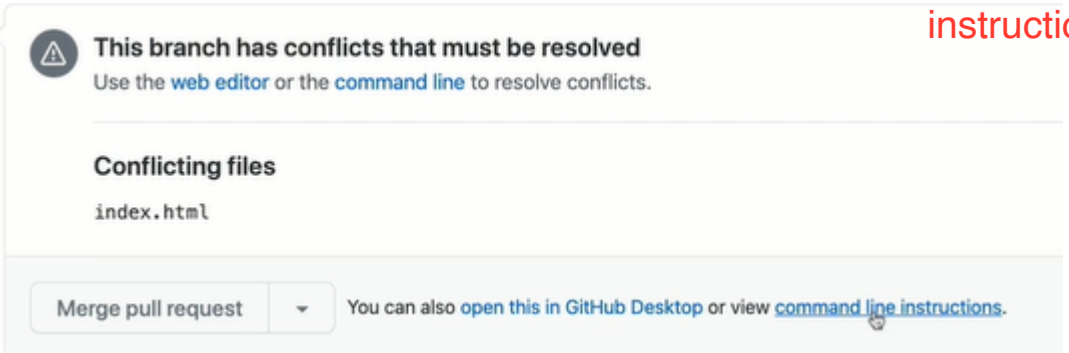






1. You can resolve the conlfict on the browser
2. You can resolve using vs code so click on command-line instructions

Or



Checkout via command line

If you cannot merge a pull request automatically here, you have the option of checking it out via command line to resolve conflicts and perform a manual merge.

HTTPS | Git | Patch | https://github.com/Colt/feature-branch-workflow.git

Step 1: From your project repository, bring in the changes and test.

```
git fetch origin
git checkout -b new-heading origin/new-heading
git merge main
```

Step 2: Merge the changes and update on GitHub.

```
git checkout main
git merge --no-ff new-heading
git push origin main
```

git fetch origin
to bring the changes that our remote repo to our local repo

```
feature-branch-workflow ❯ git switch new-heading
```

new-heading —> origin/new-heading

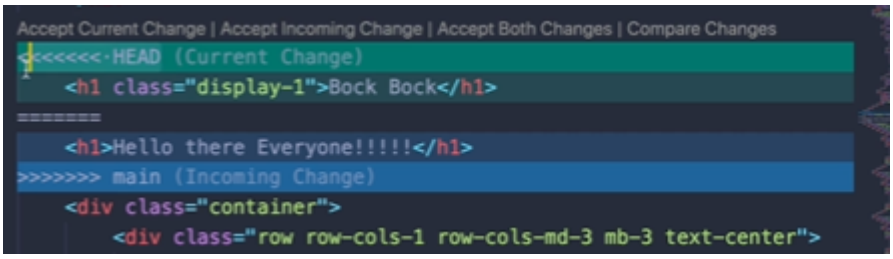
```
feature-branch-workflow ❯ git merge main      new-heading
Already up to date.
feature-branch-workflow ❯ git switch main     new-heading
Switched to branch 'main'
Your branch is behind 'origin/main' by 1 commit, and can
be fast-forwarded.
  (use "git pull" to update your local branch)
```

new heading and main are the same

we used git fetch not git pull, since origin/master is ahead. so we fix it below

```
feature-branch-workflow ❯ git pull origin main
```

```
feature-branch-workflow ❯ git switch new-heading    main
Switched to branch 'new-heading'
Your branch is up to date with 'origin/new-heading'.
feature-branch-workflow ❯ git merge main    new-heading
```



above we resolve conflicts and commit the changes.

it doesnt open vs code by itself. it just gives us in this format, what is the conflict, and we are supposed to see if we wanna accept incoming change or etc.
we have to open the file ourselves.
which file if u ask, just check with git status.

```
feature-branch-workflow ❯ git status                new-heading
On branch new-heading
Your branch is ahead of 'origin/new-heading' by 2 commits
```

```
feature-branch-workflow ❯ git switch main           new-heading
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
feature-branch-workflow ❯                                   main
```

Step 2: Merge the changes and update on GitHub.

```
git checkout main
git merge --no-ff new-heading
git push origin main
```

--no--ff that's an indicator we telling git merge to not to fast foward, if it detects it can.

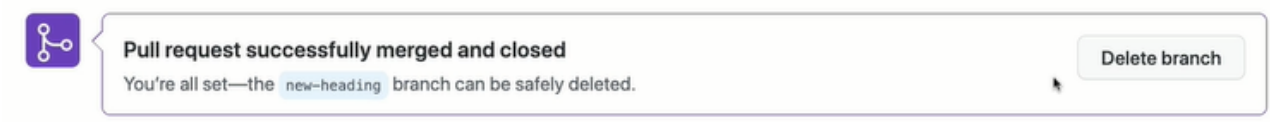
Bcoz when we do a fast foward, instead of making merge commit, it will move the branch pointer to some new commit. Sometimes you wanna prevent that from happening especially when we want to maintain a specific branch.

We want to make it clear that there was a branch merged in here just to have that history.

You dont have to do this part, but it s what github recommends.

Above , made 100% sense kazim.

```
feature-branch-workflow ❯ git merge --no-ff new-heading
Merge made by the 'recursive' strategy.
 index.html | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
feature-branch-workflow ❯ git push origin main          main
```



My Boss's Local Machine

My boss can merge the branch and resolve the conflicts locally...

Switch to the branch in question. Merge in master and resolve the conflicts.

```
> git fetch origin
> git switch my-new-feature
> git merge master
> fix conflicts!
```

Switch to master. Merge in the feature branch (now with no conflicts). Push changes up to Github.

```
> git switch master
> git merge my-new-feature
> git push origin master
```

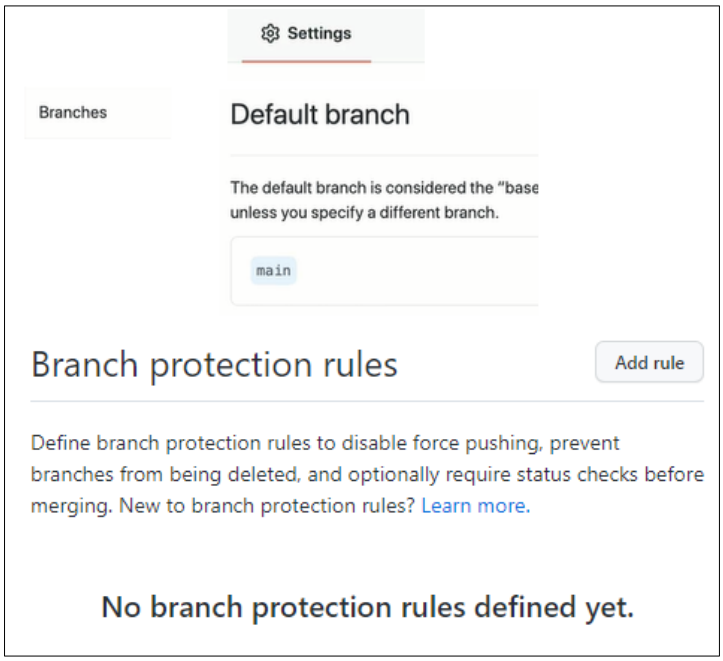
```
feature-branch-workflow ❯ git branch    main
feature-branch-workflow ❯ git branch -D new-h
eading
```

```
feature-branch-workflow ❯ git pull origin mai
n
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done
```

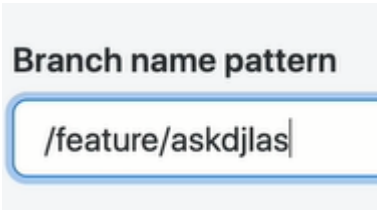
Configuring Branch protection Rules

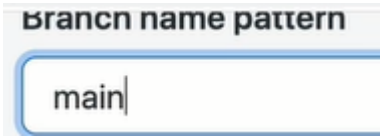
Goto settings, if you are somebody who have access to setting on a repository.

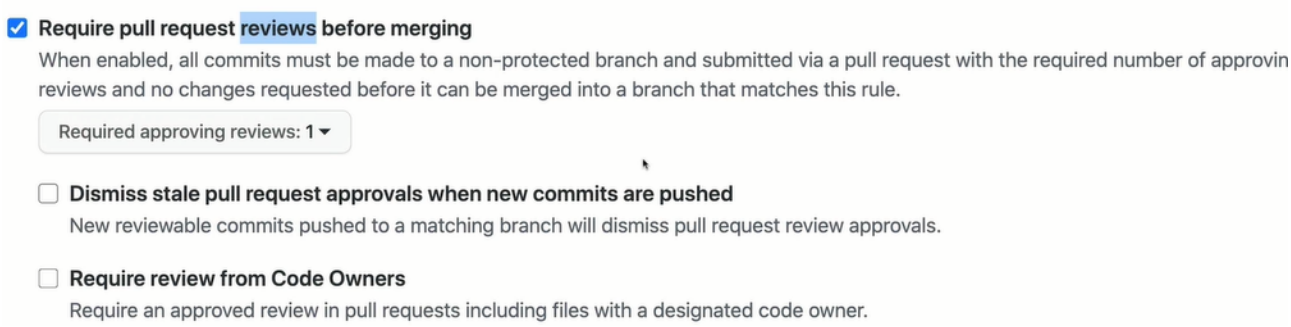
Lets say you have main and master, if you want main to be “source of truth/ official branch”.



Since, there can be 1000s of branches. Organization enforce rules.

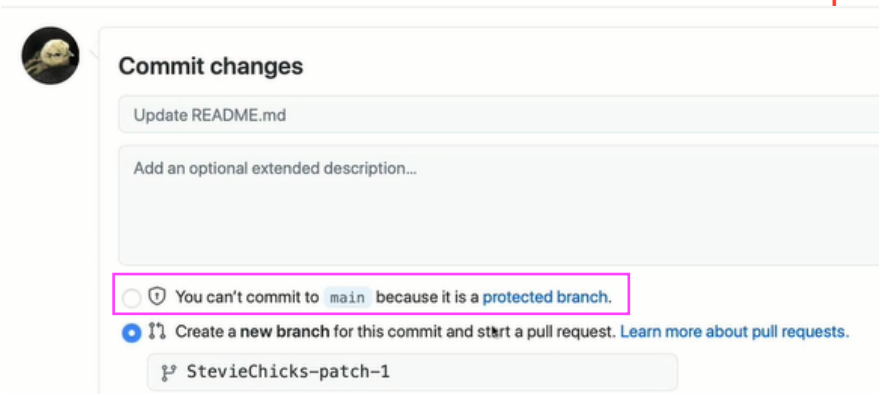




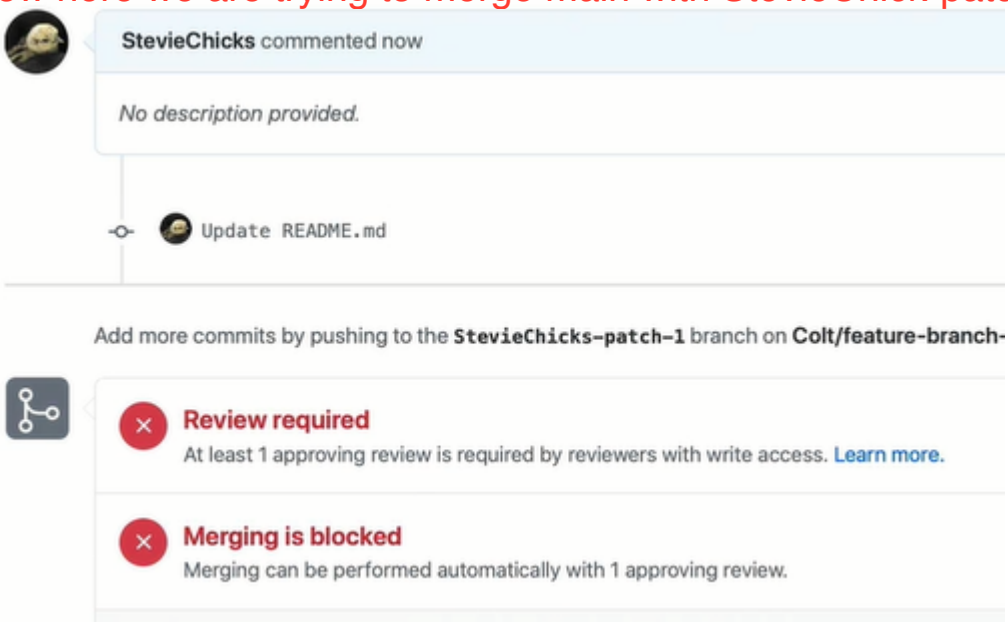


This means that our commits need to be made to a non-protected branch, basically not main and submitted via pull request and atleast one person need to approve the PR before it can be merged

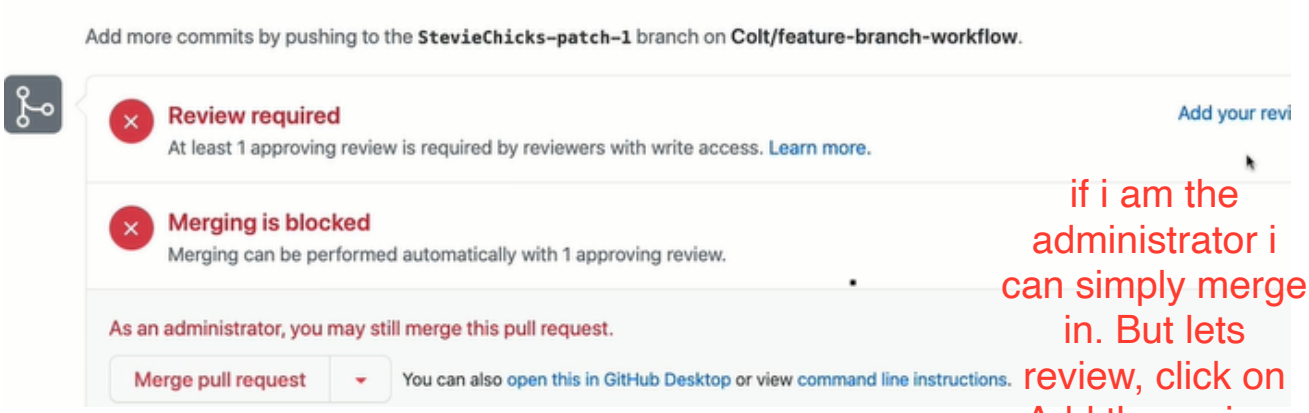
here we cant commit to main because it is a protected branch. you have to create a new branch for this commit and start a pull request.



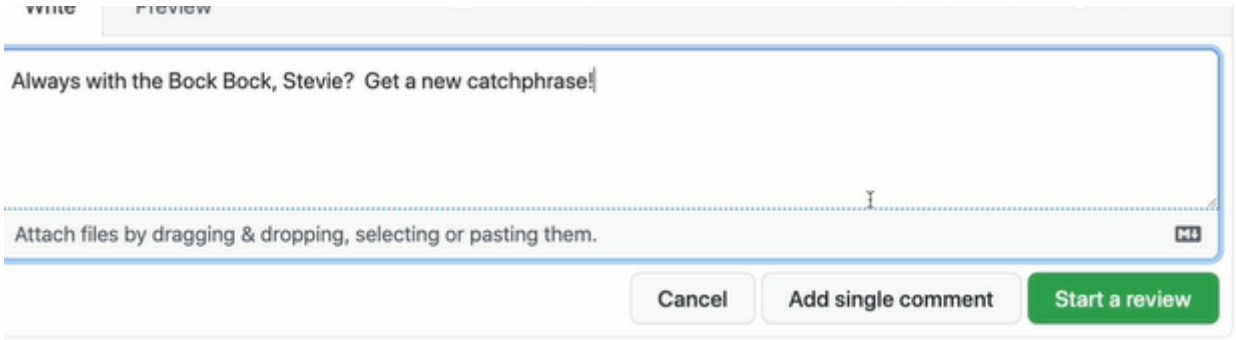
Now here we are trying to merge main with StevieChick-patch-1



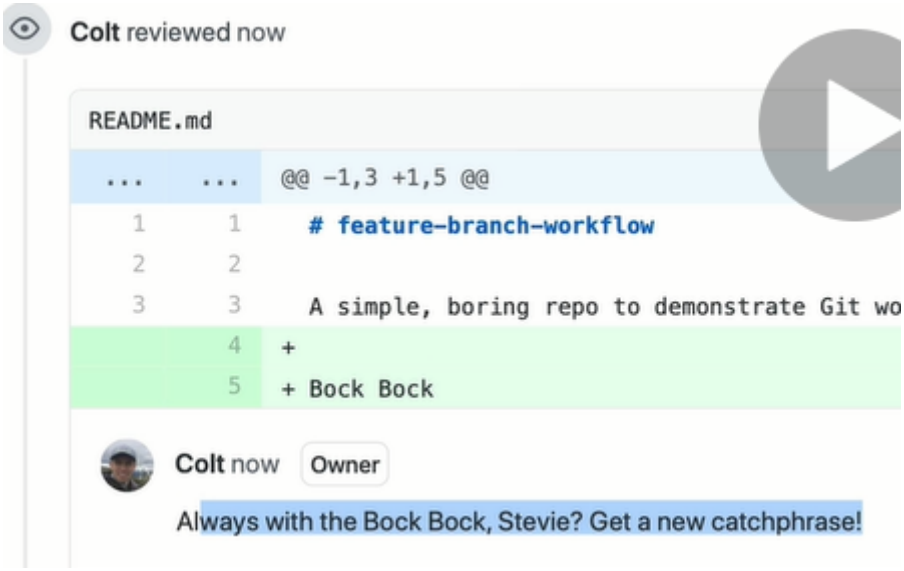
Now colt is here.

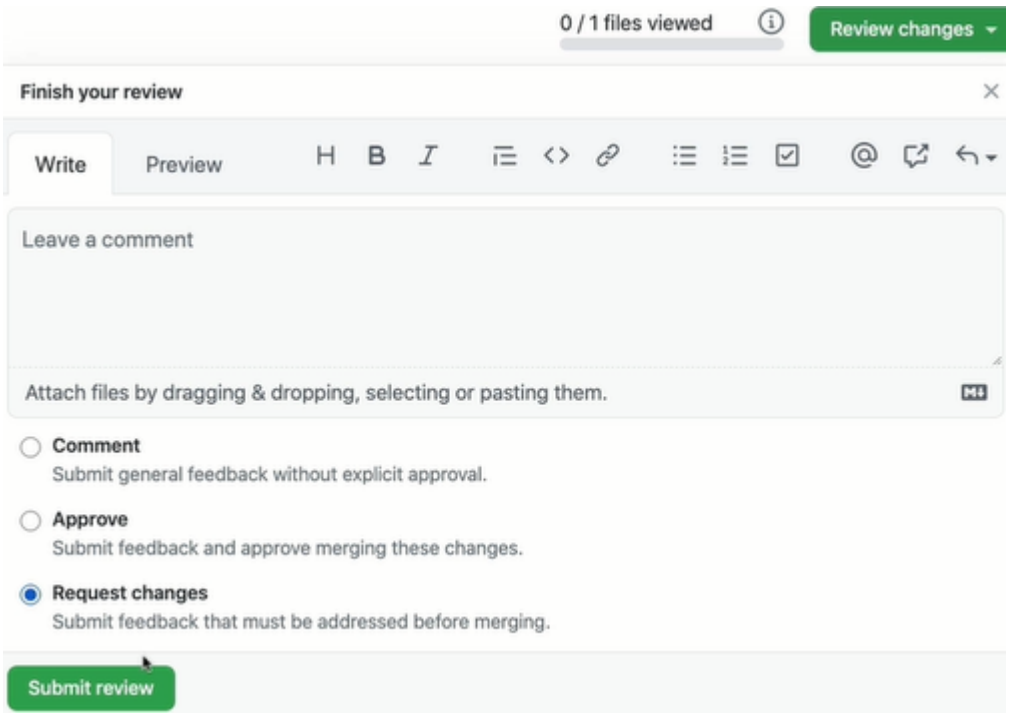


if i am the administrator i can simply merge in. But lets review, click on Add the review

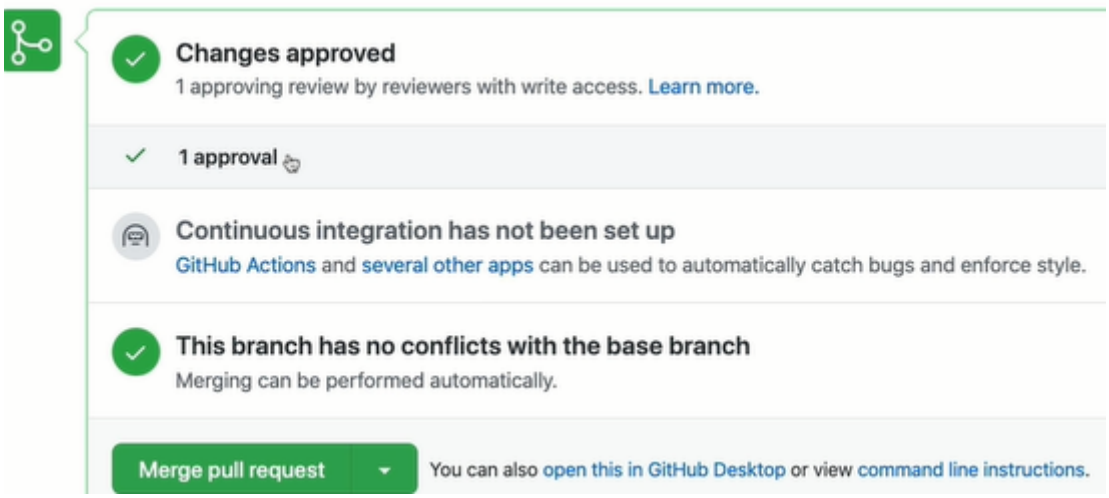


Now, stevie is checking the review done by colt





Note u can edit the file by clicking on the pencil icon. and then if u r on the main branch. u might not be able to do it.
Cuz of branch protection rules.
so u can create feature branch.
Then we do a PR.
We can see our PR being registered. But we cant merge that in. So steview will wait for colt and have a discussion with the team.



It was just 1 person is approved. Its not merge. There is even more fancy permission if we have organization setup or enterprise account.

For the PR reviewer/approval giver, he first needs to review and then he can merge.