
Let's Git This Party Started!



A Gentle Introduction to Git and GitHub

Mark Bennett, EDMUG April 2019



git

markbennett.ca



Mark Bennett, President at Burmis Studio



I'm a software developer, project manager, and meetup organizer.

I'm @MarkBennett on Twitter and GitHub.

I love talking about my kids, cooking, and travel.

Connect

<https://markbennett.ca>

@MarkBennett

Talks, social media, blog posts,
embarrassing web design.

Please hold questions

We'll stop every ten minutes for a question break, with time at the end.



Get Help

devedmonton.com

A community of hundreds of local
developers and designers.
Including EDMUG!

Slack link is at the bottom.



**Be Kind.
Be Patient.
Be Open.**



Time is
limited



markbennett.ca



I'll give you
these...



So you can do this!



**~150 slides. Guesses how far
we'll get?**

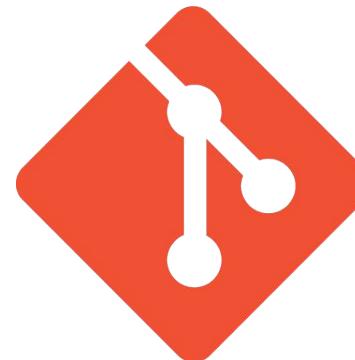


Here we go!



What is Git?

Git is a distributed version control system.



git

Ok.



But what is a version control system?

Good question!



A Version Control System (VCS) tracks and manages changes to files in a folder or project.

Remember doing a team project...



Now think of how
Google Docs made
this easy.



Version Control = Google Docs 😊

So what is a change?

Changes include things like
adding, deleting or changing
the contents of a file.

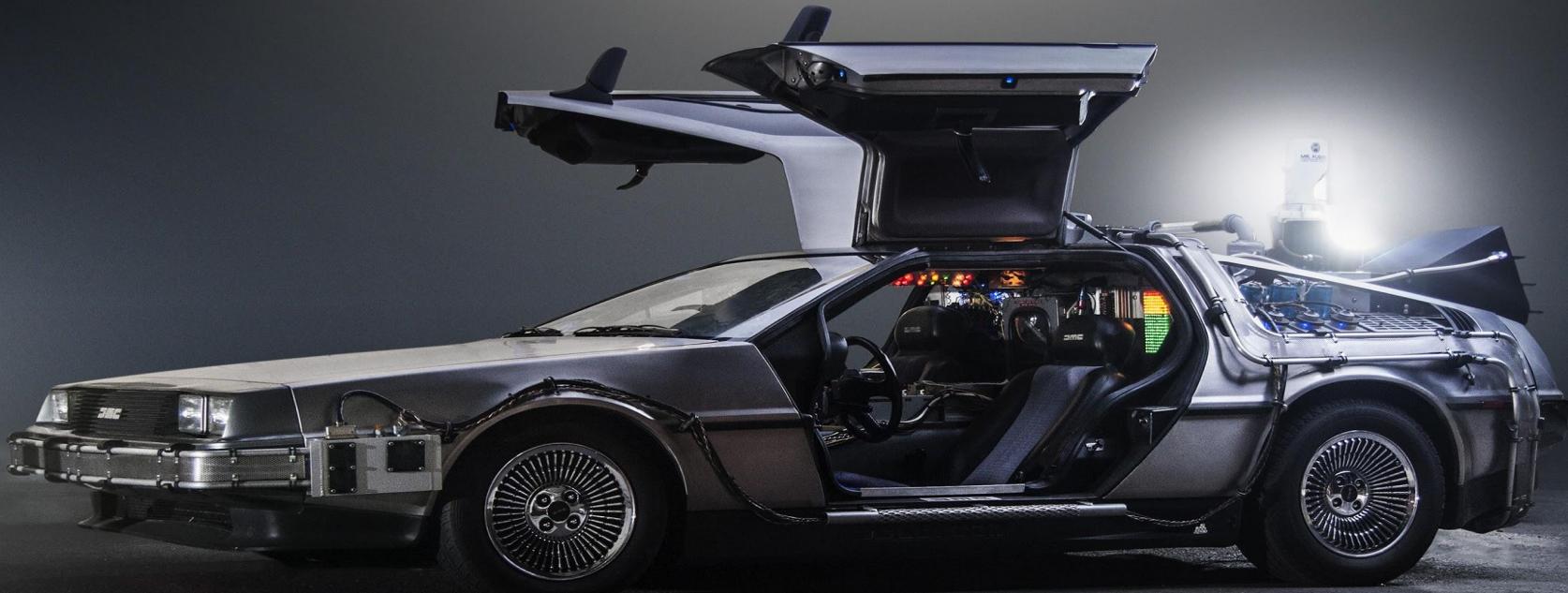
So why would I want to track and manage those changes?

Usually, it's because you want
to fix a change that broke
something. Oops.



Sometimes you just want to **try**
things out, without losing your
work.

**Version Control lets you travel
through the changes you've
made to your files, and go back
to older versions.**



It's like this! 

So what makes Git special?

Git is popular, open source,
and free.

Git also works offline.

The whole team can have their own copy of all the changes.

Git is fast and handles tricky changes well.*

* Moving and renaming files, or merging changes across branches. More later. markbennett.ca



Git makes sharing changes
easy.

What can I keep in Git?

Almost anything!

- Program source code
 - Open data
 - Images & art
 - Term papers
 - Legislation (really!)
-

So what is GitHub?

**GitHub is a
web-based hosting
service for version
control using Git.**



* With a cute mascot, called OctoCat! Check out <https://octodex.github.com>



markbennett.ca

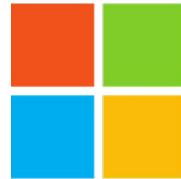


GitHub also has basic project management, code review, and documentation tools.

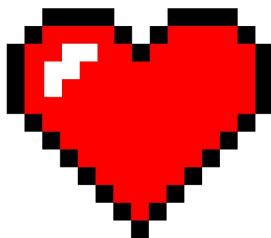
GitHub has also become a
social network for developers.



If you use open source tools,
chances are that they're using
GitHub.



Microsoft



GitHub

Alright! I've got it.

Time for lunch right?



Just one more question...

How do I actually use git?



And so the journey begins...



PAUSE.



What have we learned so far?

- A Version Control System (VCS) tracks and manages changes in files.
- Git is a VCS that's popular, open source, free, fast and works offline.
- GitHub is a web based Git hosting service and social network with project management, code review, and documentation.

Questions?

Getting Started With Git



Git

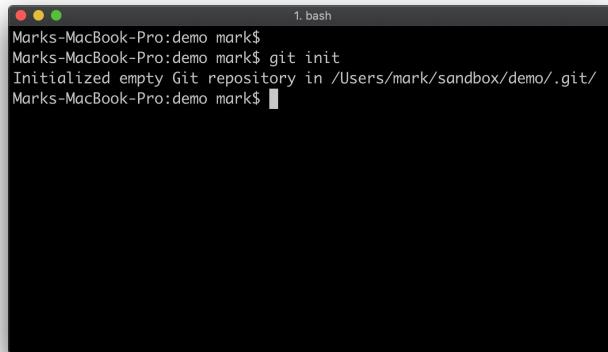
- Popular git clients
- Starting a repository
- The commit history
- Committing your changes
- Working with branches
- Merging branches

Popular Git Clients

A Git client is a tool you use to work with Git.

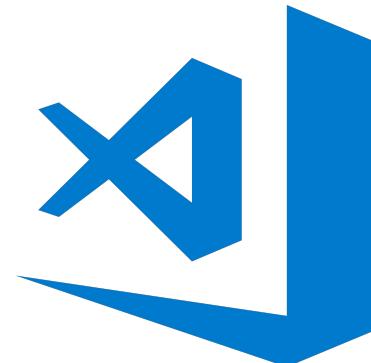
Most allow you to [create a Git repository](#), [add or change files](#), and [collaborate](#) with other people.

Popular Git Clients



```
Marks-MacBook-Pro:demo mark$  
Marks-MacBook-Pro:demo mark$ git init  
Initialized empty Git repository in /Users/mark/sandbox/demo/.git/  
Marks-MacBook-Pro:demo mark$
```

Git Command Line



Visual Studio Code



SourceTree

Popular Git Clients



I'll use [GitHub Desktop](#) for some of my examples.

Any Git client should also work.

<https://desktop.github.com>

Starting a repository

Starting a repository

Before you can track changes to your files you need to create [a repository](#).

A repository is where all the files in your project go. The repository also included the [Commit History](#) and [git settings](#). More on those later.

Starting a repository

repository = repo

Starting a repository

Most git clients keep the repository settings in hidden files and folders:

- `.git`
- `.gitignore`
- `.gitattributes`

If you see these files in a folder then you're probably looking at a Git repository. 😊

Starting a repository



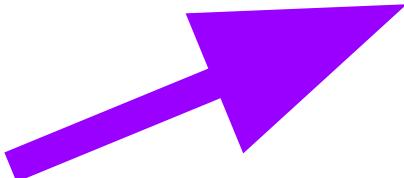
Clone a Repository from the Internet...



Create a New Repository on Your Hard Drive...



Add an Existing Repository from Your Hard Drive...



The Commit History

The Commit History

The Commit History stores snapshots, called commits, of your project from the start of the repository.

Changes History

No Branches to Compare

Explain how to get help

 Mark Bennett committed just now

Add notes

 Mark Bennett committed 41 minutes ago

Initial commit

 Mark Bennett committed 21 hours ago

The Commit History

Each commit is a snapshot of the entire project at the point you commit.*



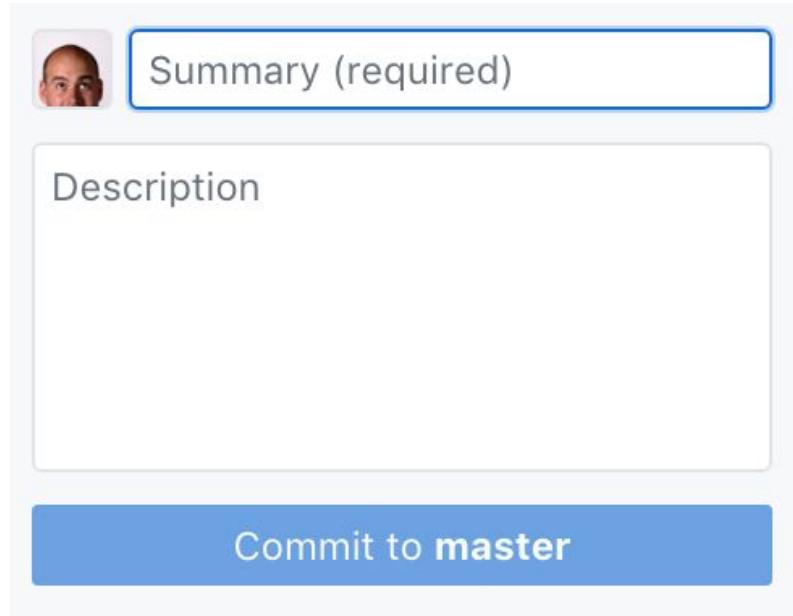
* Don't worry about space, it's compressed as you work.

The Commit History

Think of the Commit History as the story of your repository.

Each commit has a summary and description.

These tell how your commit fits into the story of the repository's history.



The Commit History

Git calculates a special code, called the hash, from the entire snapshot of a commit.

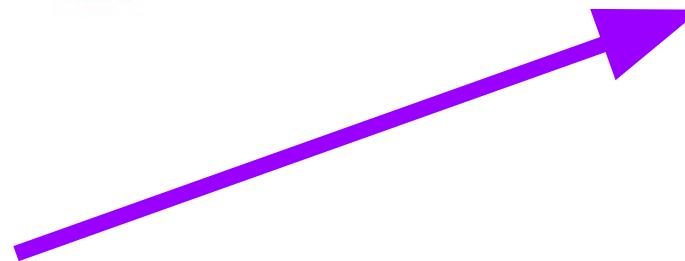
Each commit is identified by a hash.

Changing even 1 bit of the contents
and the entire hash changes.

Explain how to get help



Mark Bennett committed  0f15e22



Committing Your Changes

Committing Your Changes

You need to commit changes to the history manually.

They're not saved automatically.

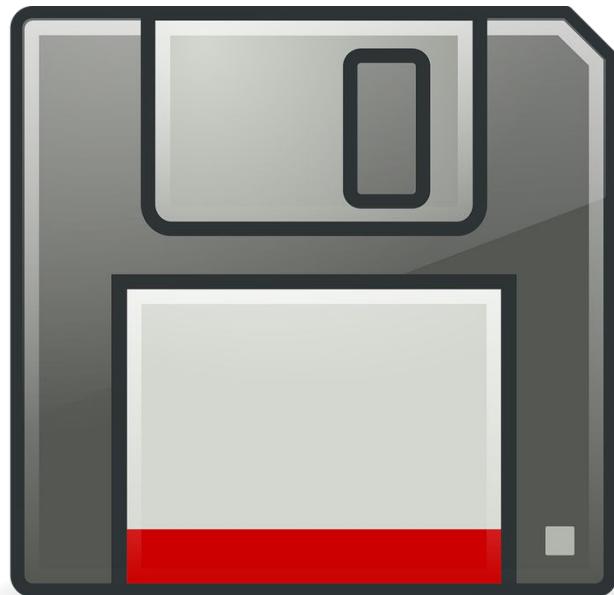


Committing Your Changes

1. Make the changes
2. Stage the changes
3. Commit the changes

Committing Your Changes

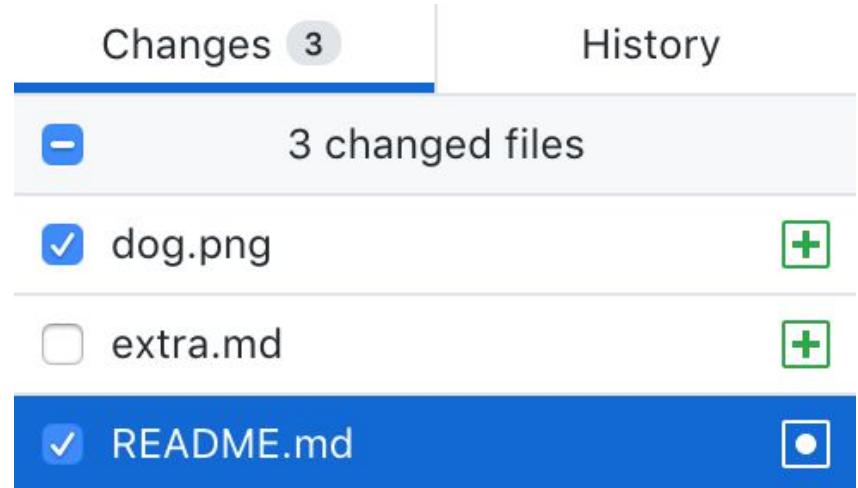
Make the changes in your editor or application then save them to the disk.



Committing Your Changes

Stage the changes you'd like to add to the next commit.

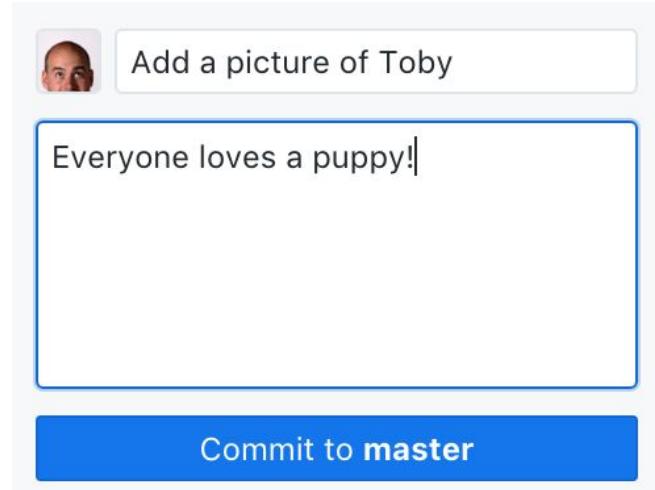
You don't need to stage all the changes you've made, and you can stage or unstage them until you're happy.

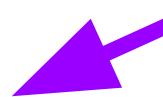


Committing Your Changes

Once your changes are staged,
[review them.](#)

If you're satisfied add the summary
(and description) then [commit the](#)
[changes to the repo!](#)





Committing Your Changes

Your changes now appear as a new commit in the Commit History.

Congratulations!

Add a picture of Toby



Mark Bennett committed just now

Explain how to get help



Mark Bennett committed 11 hours ago

Add notes



Mark Bennett committed 11 hours ago

Initial commit



Mark Bennett committed a day ago

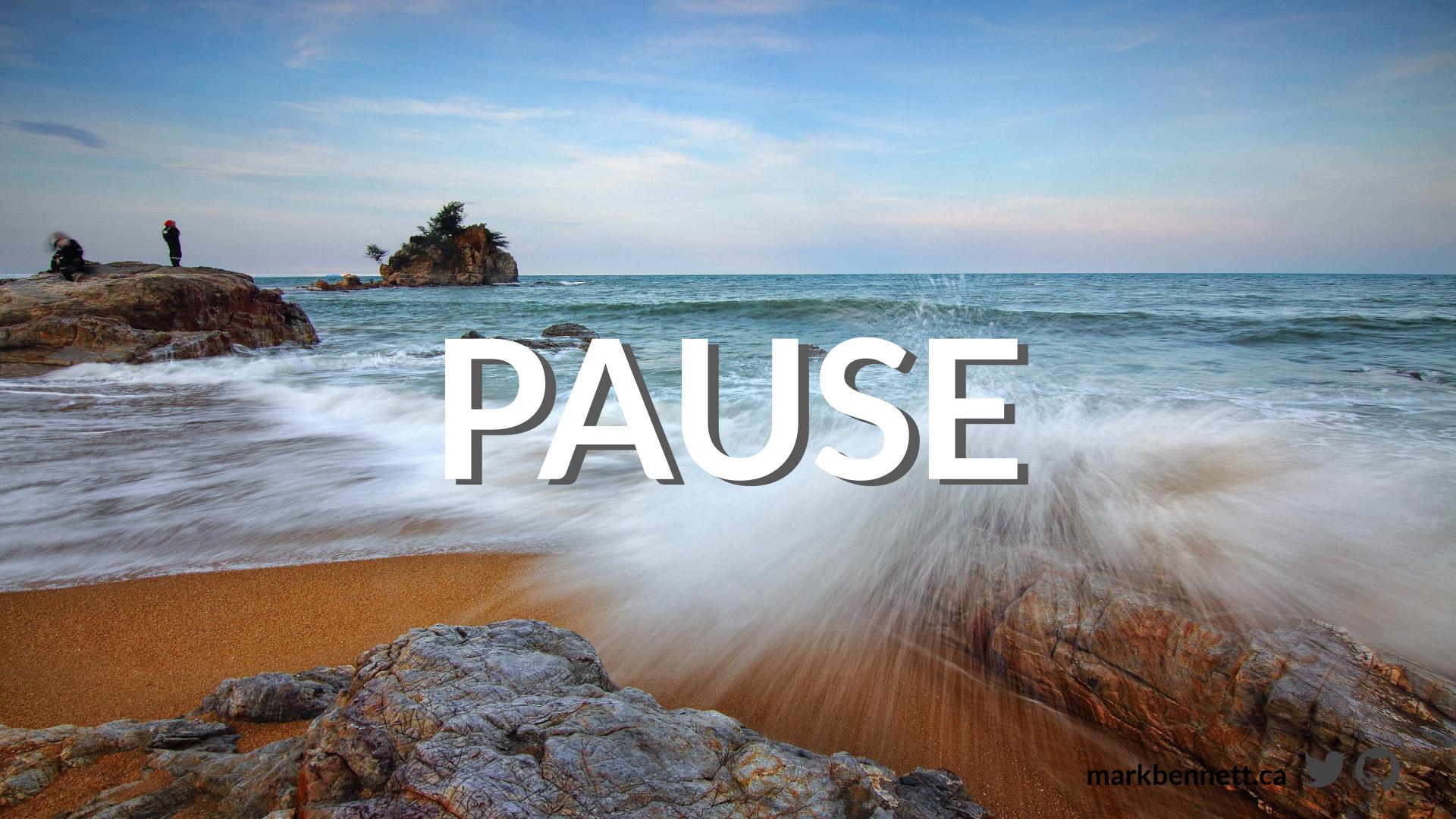
Committing Your Changes

Try to remember you're telling a story you might care about later.

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Comic courtesy of XKCD.

A wide-angle photograph of a coastal scene at sunset. The sky is a gradient from blue to orange and yellow. In the foreground, there are large, textured rocks on the left and right. Waves are crashing onto a sandy beach in the center. On the left side, two people are standing on a rocky outcrop. In the middle ground, there is a small, isolated rock formation with a tree growing on it. The ocean extends to the horizon.

PAUSE

Review

- [Create a repository](#) to start using Git.
- A [commit is like a snapshot](#) of the whole project.
- We edit, stage, then commit changes.
- The [commit history tells the story](#) of your project.

Questions?

**If each commit is snapshot of
the whole repository, why isn't
my Git folder huge?**

You've been paying attention!

Git uses compression and some advanced data structures so that only the differences between commits are stored.

Because of this it uses much less space than if it kept a fresh copy every time.

Working With Branches



Working With Branches

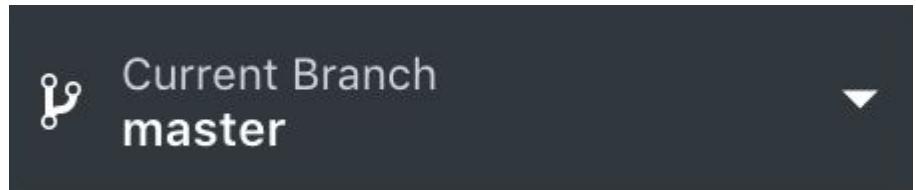
Sometimes you want to work on a new feature, fix a bug, or just experiment.

Git branches let you do this safely and easily.

Working With Branches

You've actually got a branch already!

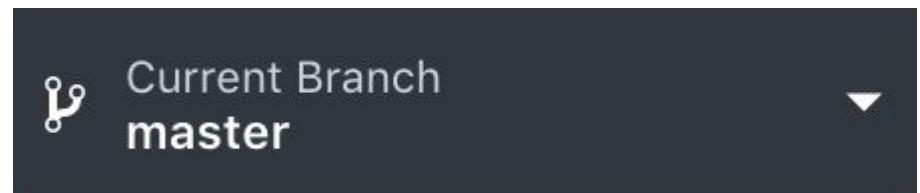
It's called the master branch and is made for you when you start your repository.



Working With Branches

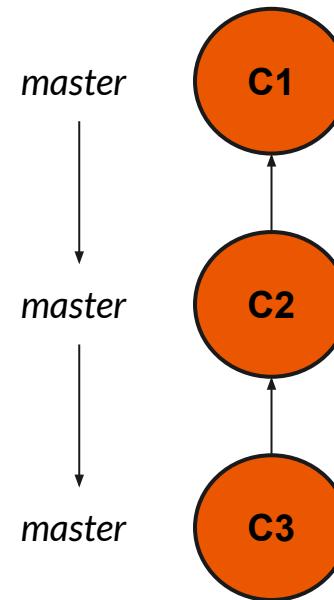
By default, “*master*” is the currently checked out branch.

The checked out branch is where all your commits will go as you work.



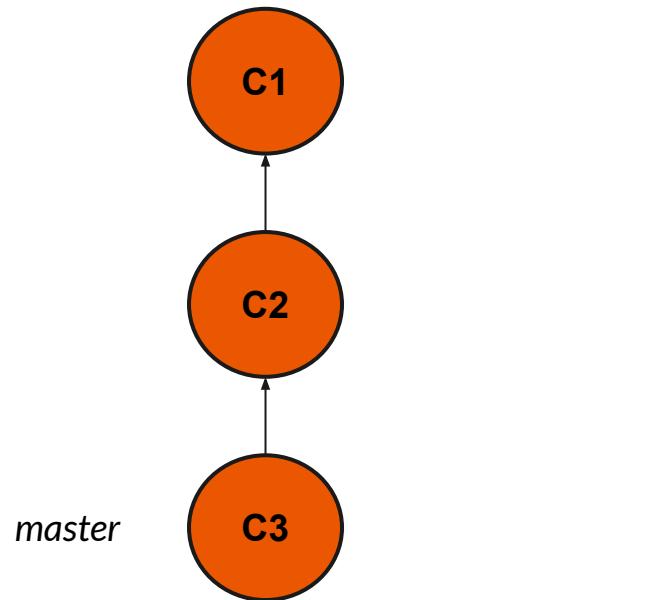
Working With Branches

As you make commits, the “master” branch automatically points to the latest commit you’ve made.



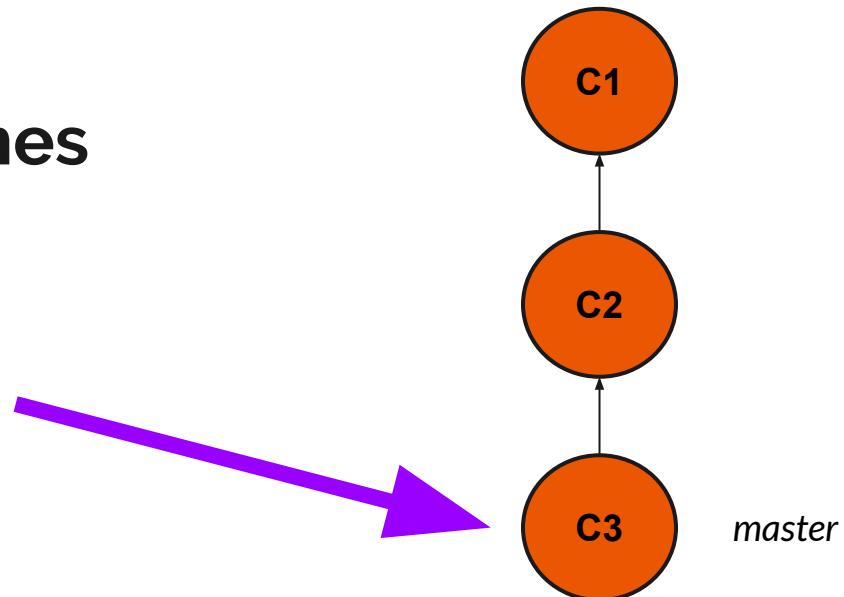
Working With Branches

A branch is like a special pointer to a commit which changes as you work.



Working With Branches

Here “*master*” points to the third commit (C3).



Working With Branches

Making new branches and checking them out is easy!

Create a Branch



Name

bugFix

Your new branch will be based on your currently checked out branch (`master`). `master` is the [default branch](#) for your repository.

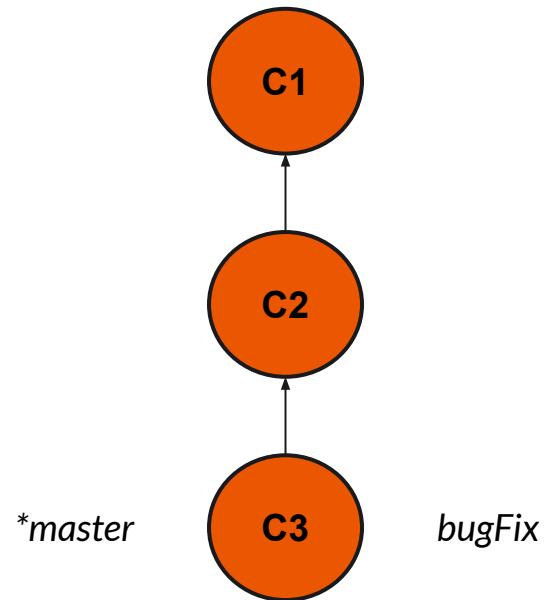
Cancel

Create Branch

Working With Branches

You branch off an existing commit.

Here both “*master*” and “*bugFix*” point to the same commit.

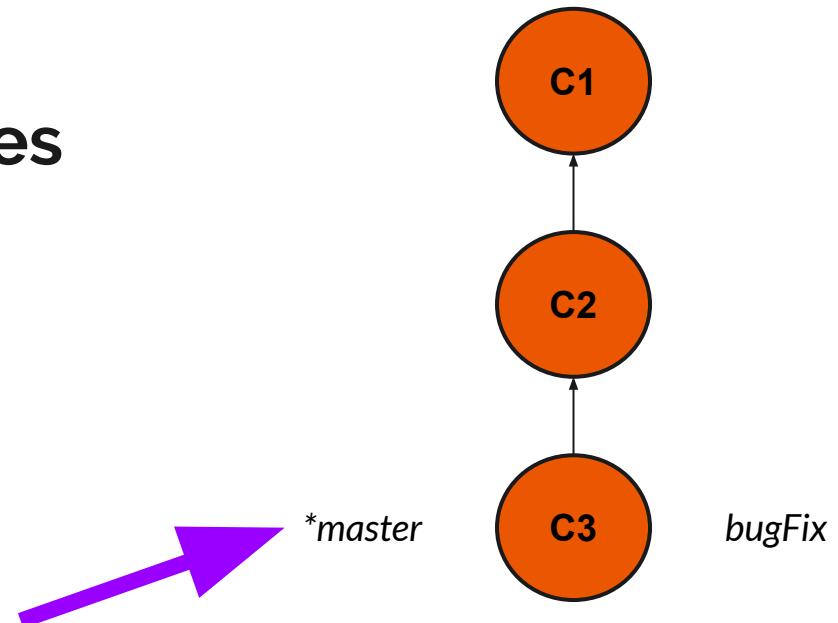


Working With Branches

You can only checkout one branch at a time.

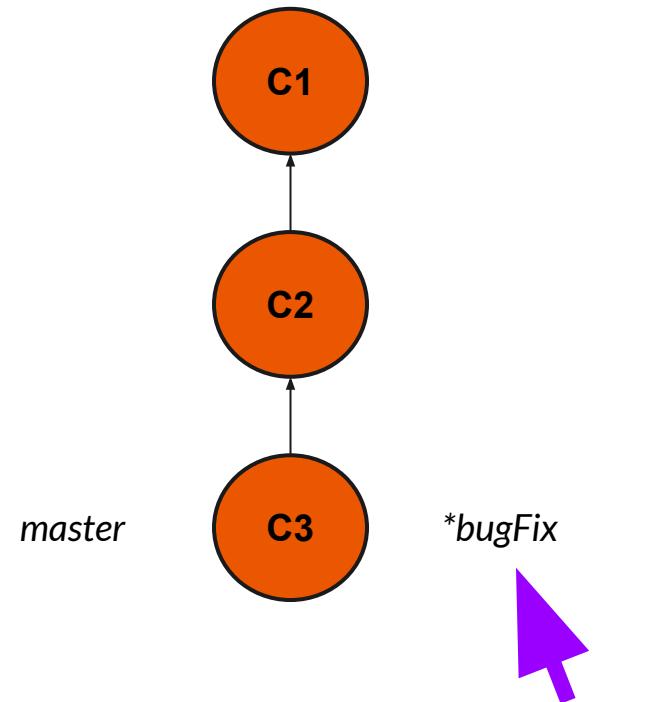
We'll use a "*" to indicate which branch is checked out.

Here's we've checked out "master".



Working With Branches

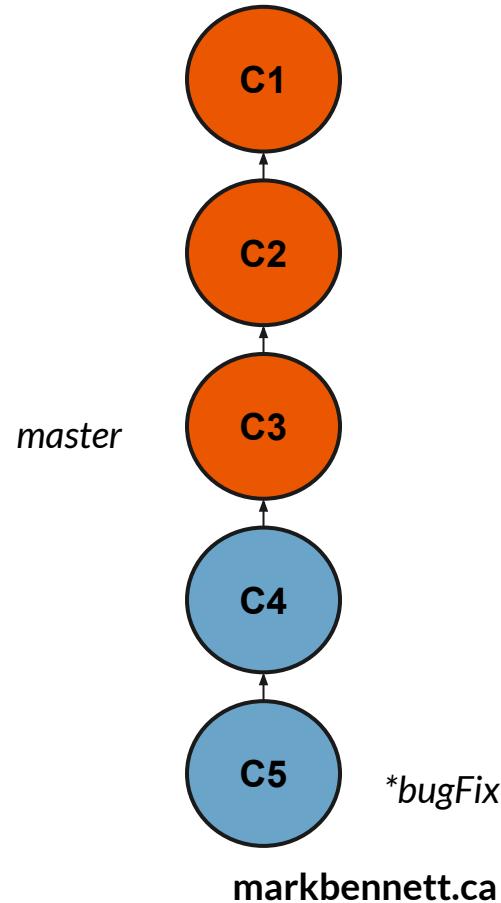
Now we've checked out “*bugFix*” instead.



Working With Branches

As we make commits “*bugFix*” will move.

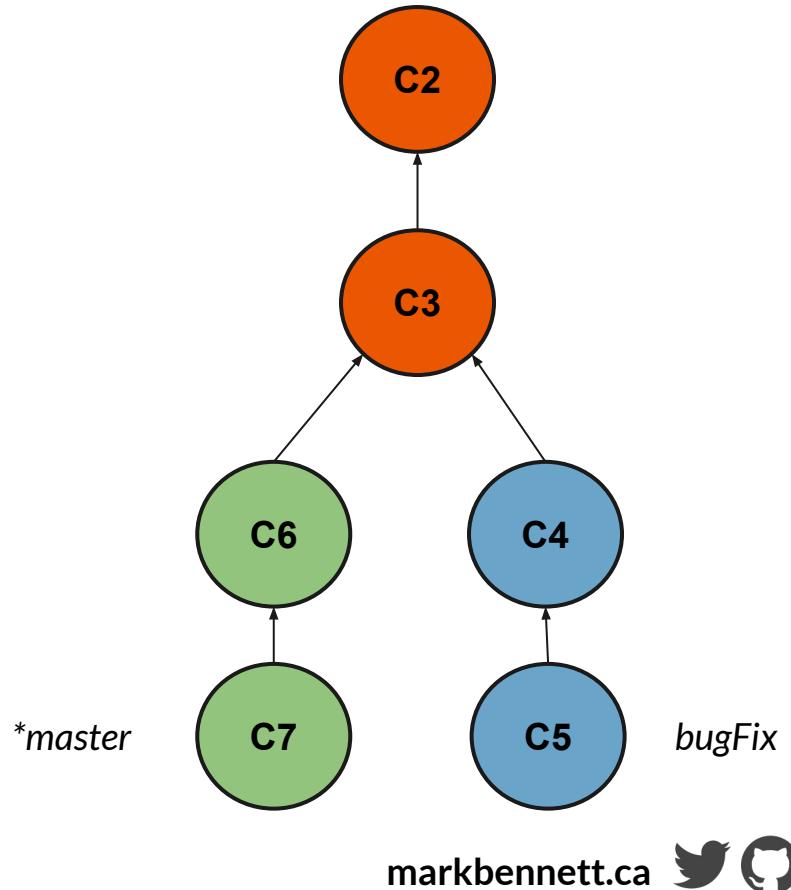
“*master*” still points to our old commit.



Working With Branches

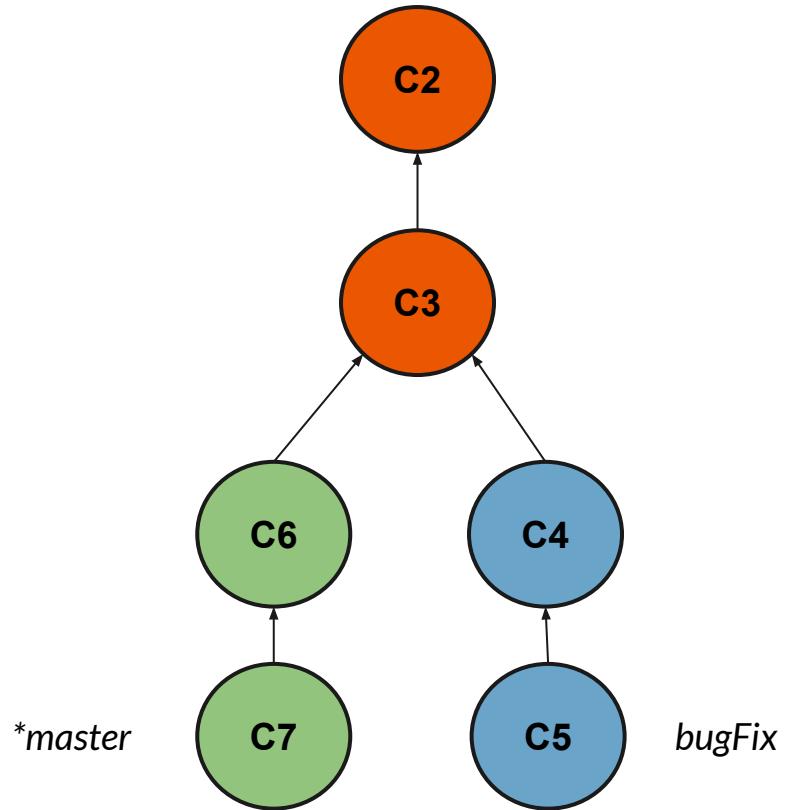
If we checkout “master” again and make some commits, something interesting happens!

Note I've cutoff (C1) to save space.



Working With Branches

We've got commits with the same parent, but with different changes after!



How do we get our bug fix back into our project?

We'll answer in a moment!



PAUSE

Review

- Branches let you work and commit independently from one another.
- Branches are used for bug fixes, experiments, new features, work in progress.
- New commits are added to the branch you've checked out.
- You always have one branch checked out.
- By default every repository has a “*master*” branch.

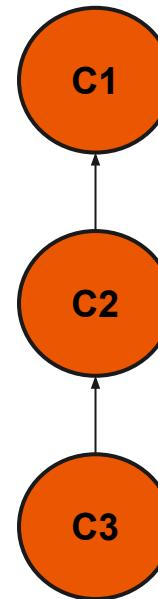
Questions?

Merging Branches

Merging Branches

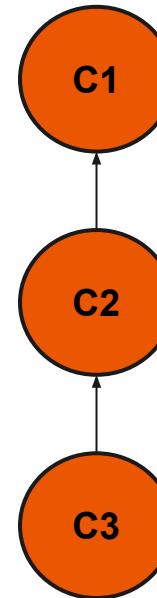
Before you understand merging branches you must understand commits and the commit history.

Each commit has a parent.



Merging Branches

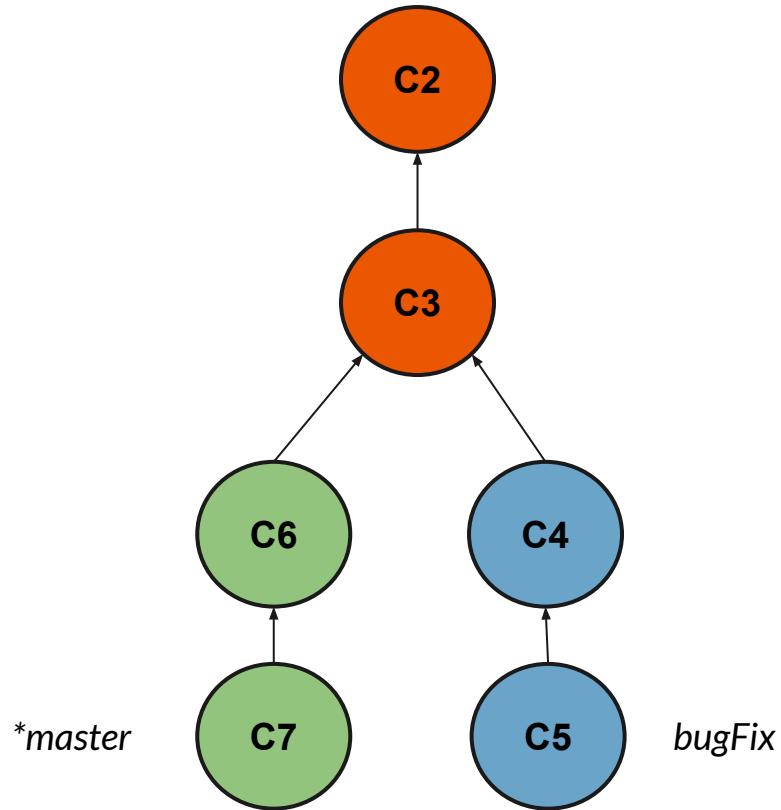
We've talked about branches as pointers, but a branch also says that, "I want to include the work of this commit and all its parents".



Merging Branches

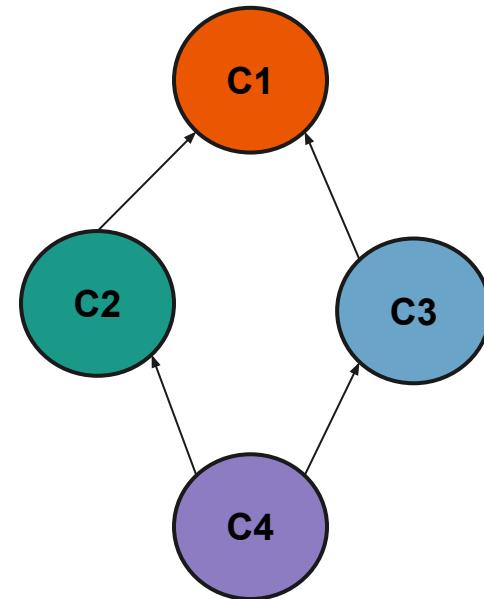
“master” includes C2, C3, C6, C7

“bugFix” includes C2, C3, C4, C5



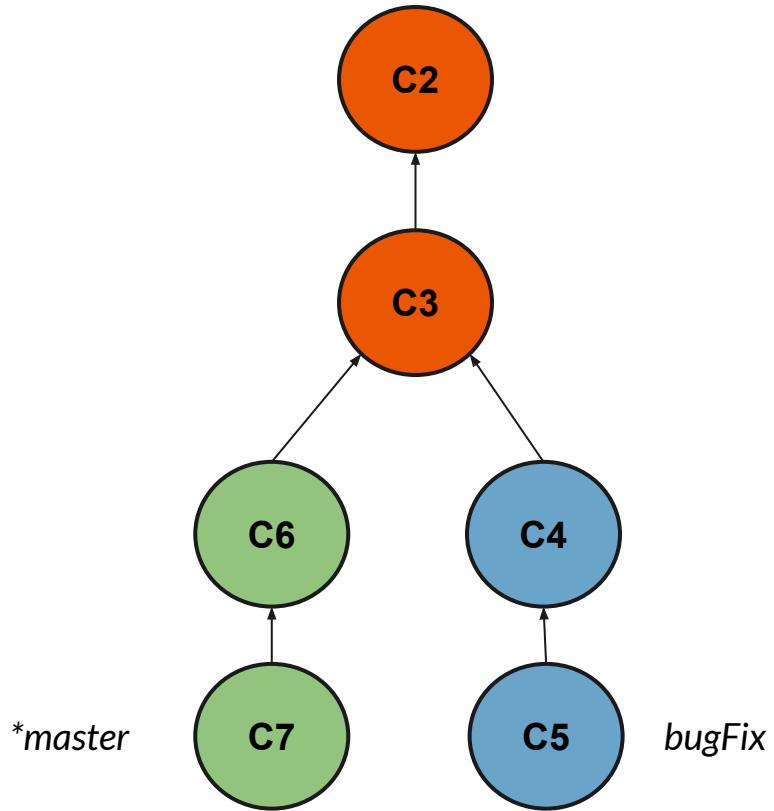
Merging Branches

Each commit can actually have zero (C1), one (C2 and C3), or many parents (C4)!



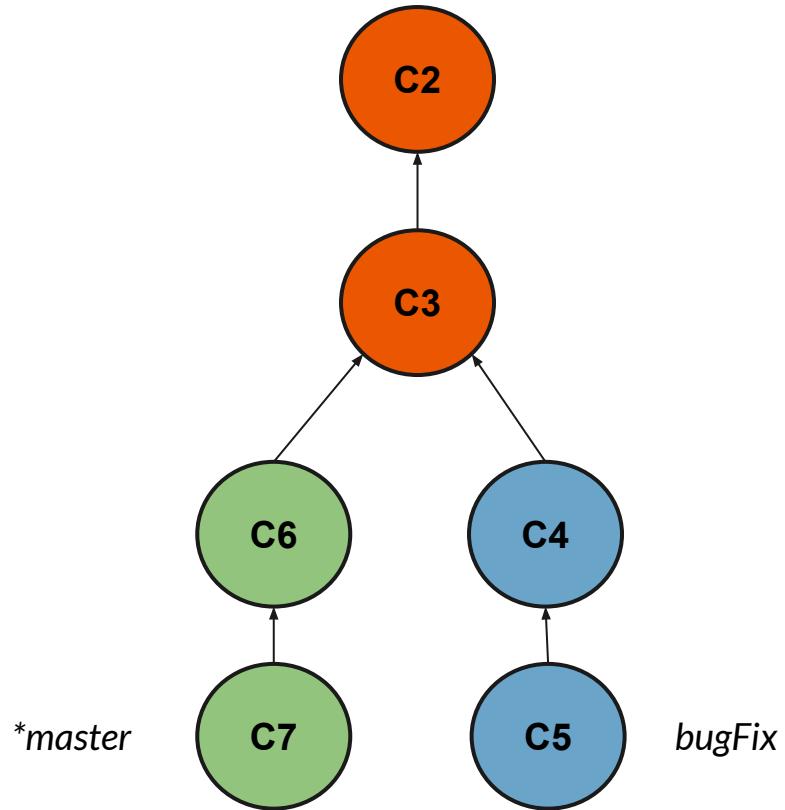
Merging Branches

How do we get our bug fix back into our project?



Merging Branches

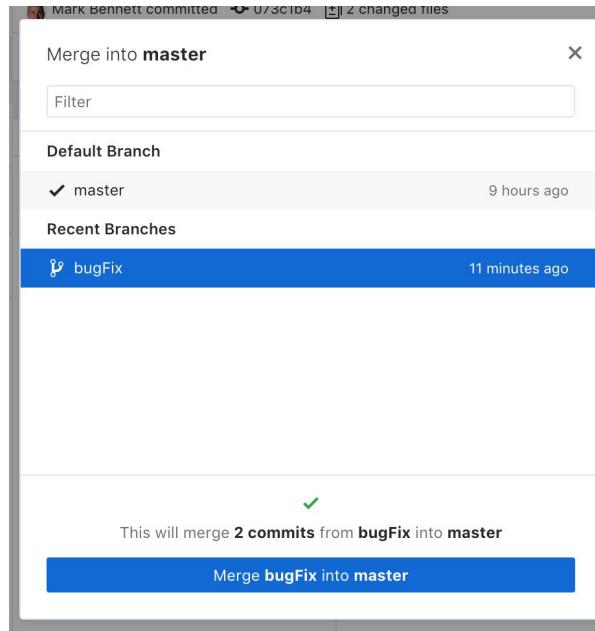
We can get the changes from
“*bugFix*” into “*master*” by merging.



Merging Branches

Git understands which work is in each branch.

It can calculate and apply the differences between the branch commits.



Merging Branches

Each merge creates a new commit in the commit history.

Here we merged “*bugFix*” into “*master*”.

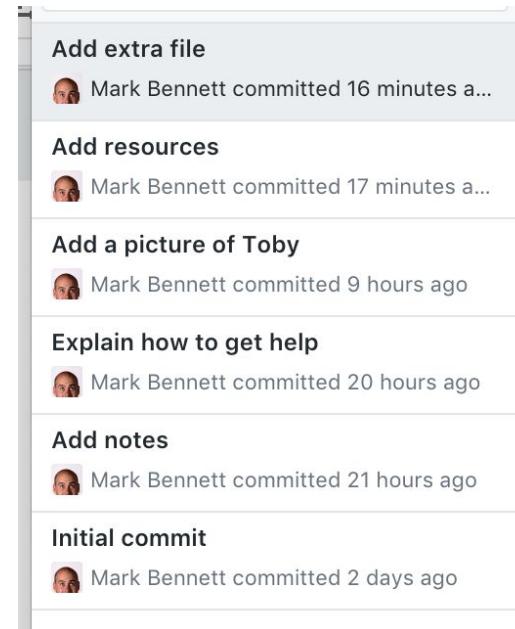
You can see the commits from “*bugFix*” in “*master*” now”

Add extra file
Mark Bennett committed 16 minutes ago
Add resources
Mark Bennett committed 17 minutes ago
Add a picture of Toby
Mark Bennett committed 9 hours ago
Explain how to get help
Mark Bennett committed 20 hours ago
Add notes
Mark Bennett committed 21 hours ago
Initial commit
Mark Bennett committed 2 days ago

Merging Branches

Note that by default GitHub Desktop and many other tools don't show merge commits.

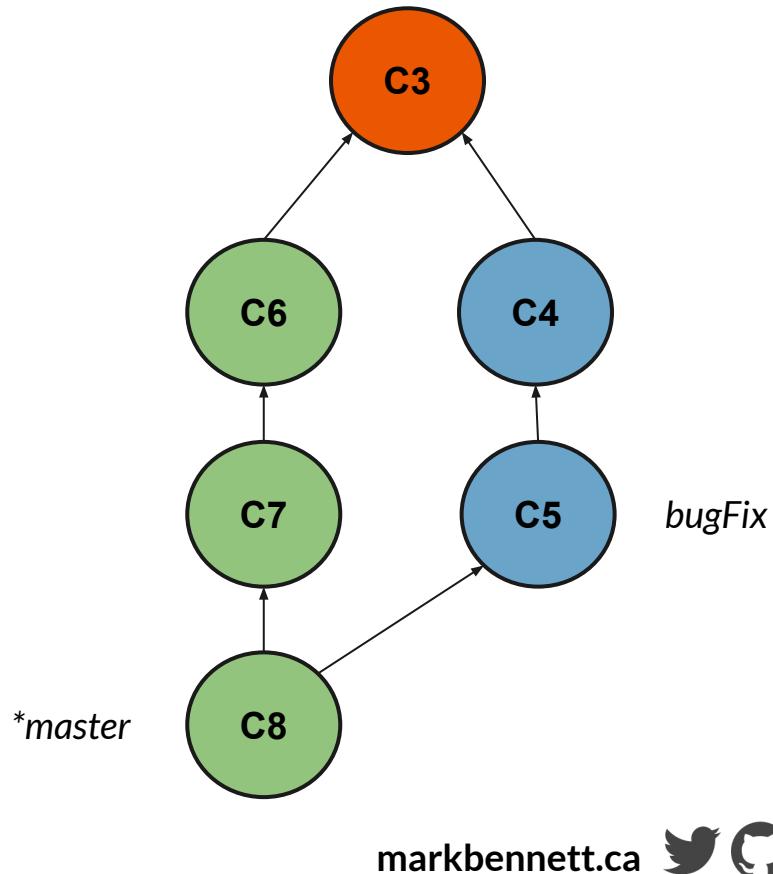
You can use the command-line and `git log` to see merge commits.



Merging Branches

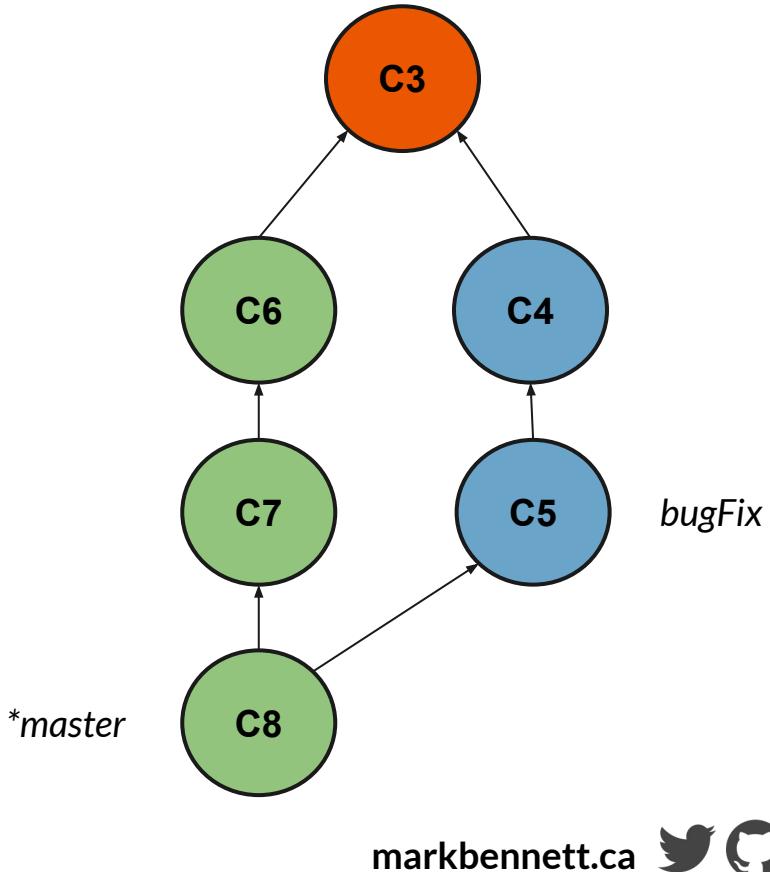
After the merge our “*master*” branch points to a new commit (C8) with the work from “*bugFix*” included.

This commit has two parents (C5 and C7)



Merging Branches

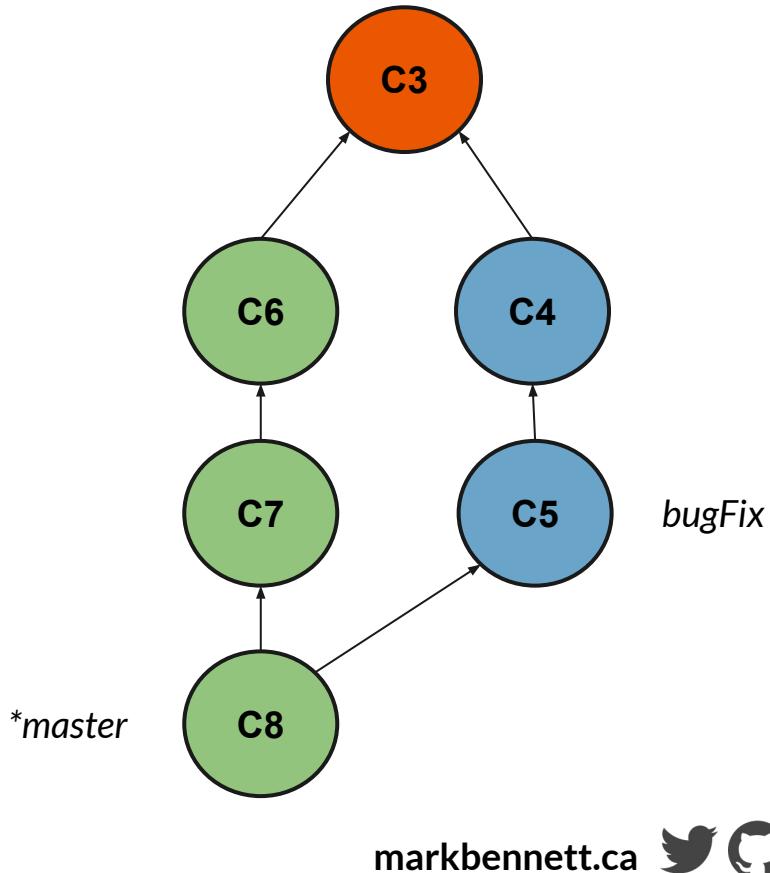
Sometimes a merge fails because changes you're merging in would conflict with changes in your branch.



Merging Branches

When there is a conflict, your Git client will stage all the changes it can automatically.

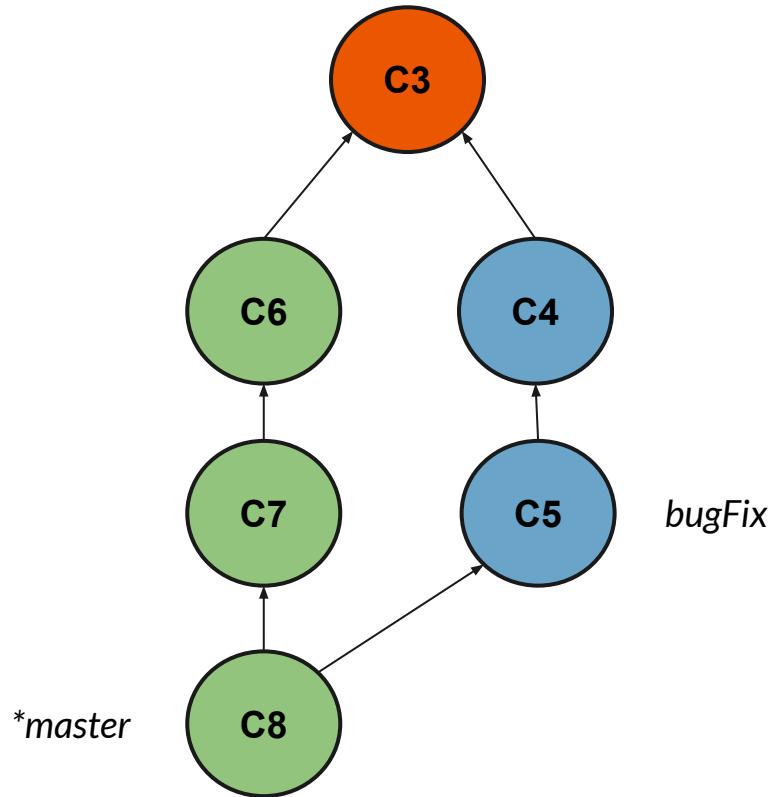
Both copies of the conflicting changes will also be included.



Merging Branches

You must manually edit the files
and stage these changes to resolve
the conflict.

Once it's resolved you can commit
the staged changes and the merge
is done!





PAUSE

Review

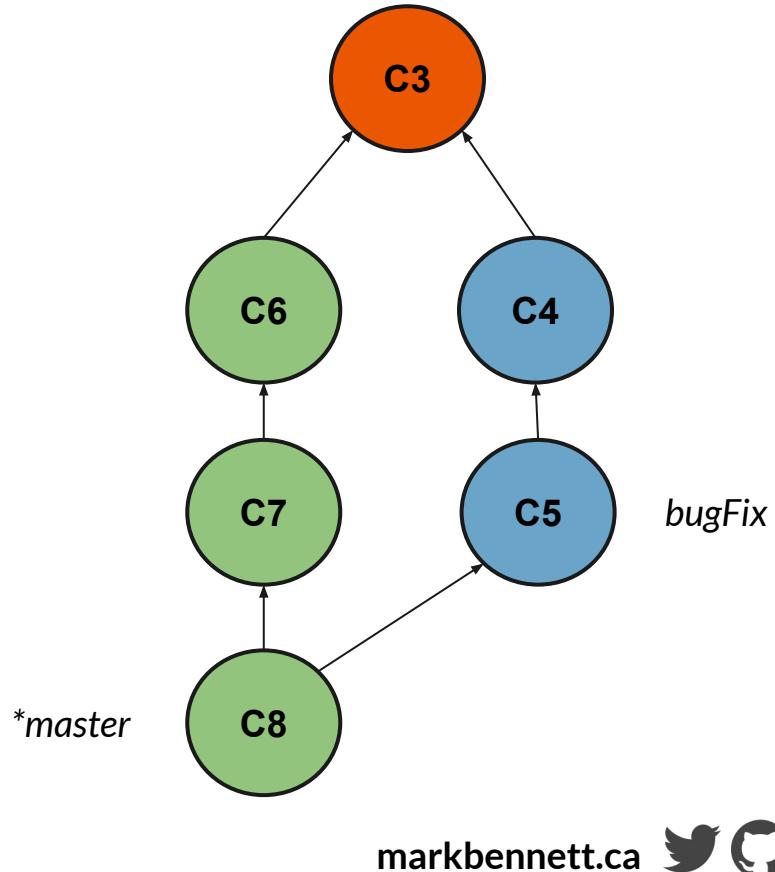
- Branches point to a commit.
- Branches track all the commits included in them.
- Commits can have more than one parent.
- Merging branches makes a new commit with two parents.
- Merging is usually automatic, but conflicts must be fixed manually then committed.

Questions?

Keeping The Git Log Clear

`git merge` is safe but can result in a confusing log when a commit has more than one parent.

Git offers a tool called `git rebase` you can use to edit your commit log to make it cleaner before sharing.



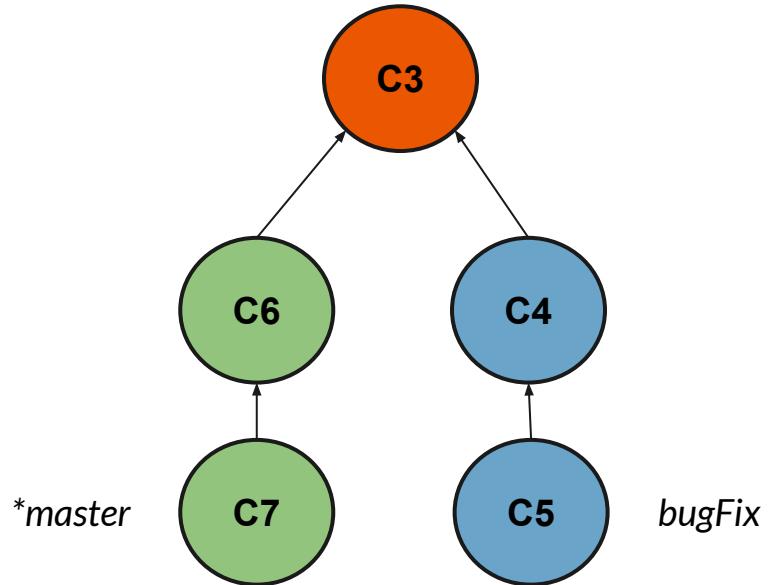
Git Rebase

By rebasing you take the commits in one branch and **create new commits by replaying the changes from each commit against another branch.** The original commits are then deleted!

Typically, you rebase a feature branch against `master`.

Git Rebase

For example, imagine you've got a `bugFix` branch that you want to rebase so your changes follow the head of the `master` branch (C7)

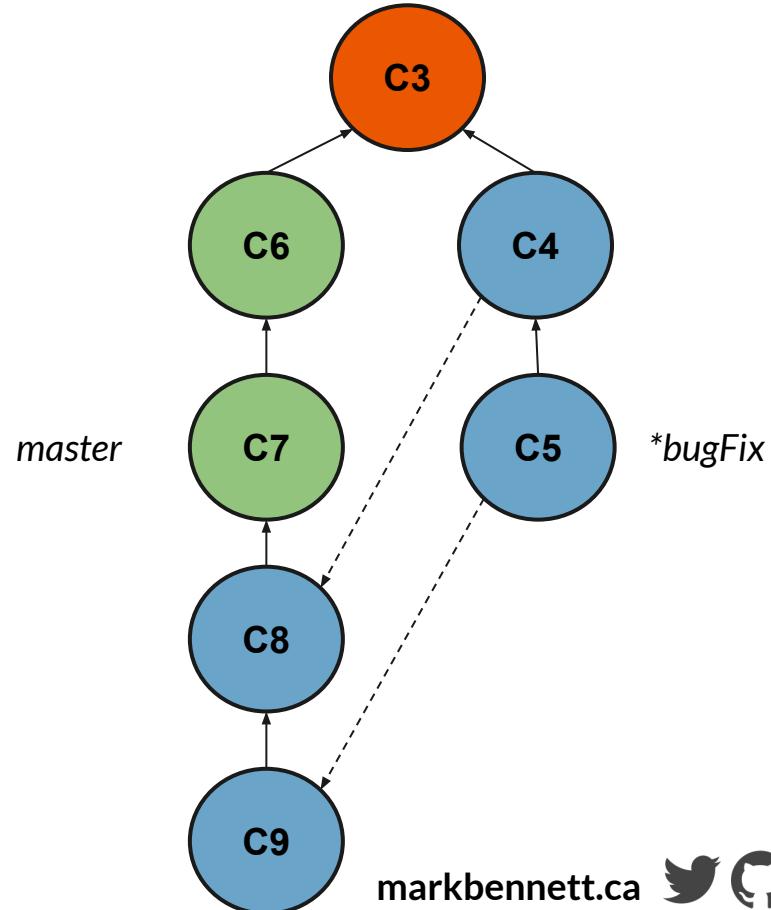


Git Rebase

If you run:

```
git checkout bugFix  
git rebase master
```

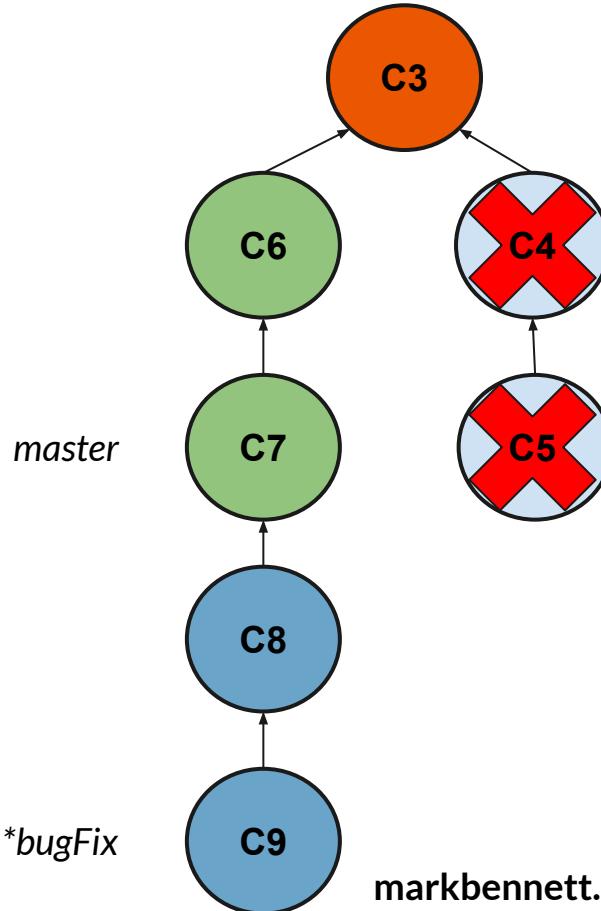
Then Git will take the changes in **C4** and apply them to **C7** to make a new commit **C8**. It then applies changes from **C5** to make **C9**.



Git Rebase

Now that the new commits have been created, the branch is updated to point to C9.

The old commits are then deleted.



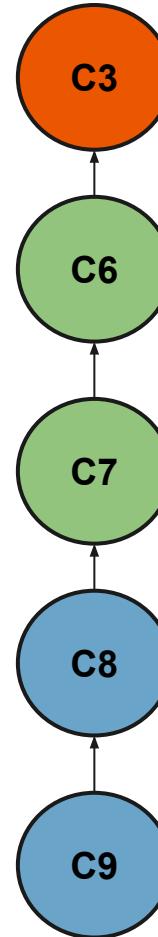
Git Rebase

If you merge the `bugFix` branch into `master`, Git doesn't need to edit the history at all.

Git updates `master` to point to the latest commit in the `bugFix` branch.*

* A merge where Git just has to move the branch to the latest commit is very easy and has a special name, a **fast-forward merge**.

**master, bugFix*



markbennett.ca



WARNING: Git Rebase

NEVER REBASE A BRANCH
YOU'VE SHARED!

`git rebase` can only delete
commits from your local branch!*

If you shared commits, then send
rebase commits the changes will be
shared twice!

* Technically, you can force push a local branch, more later.



A close-up photograph of two young dogs resting together. In the foreground, a tan-colored puppy with dark brown ears and a black-tipped tail is looking towards the right. Behind it, a black and tan dachshund puppy with a tan patch on its chest is also looking in the same direction. They are lying on a light-colored, textured surface, possibly a bedsheet or blanket.

PAUSE

Review

- Keeping your Git log clean is helpful
- `git rebase` can be a useful tool
- Using `git rebase` takes care
- NEVER REBASE COMMITS YOU HAVE SHARED!

Questions?

Getting Started With GitHub

GitHub

- Sharing your code
- Your GitHub repo
- Issues
- Contributing and Pull Requests
- Exploring GitHub

Sharing Your Code

Sharing Your Code

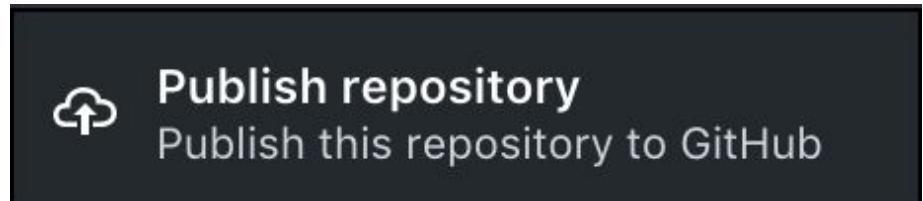
Git works great on it's own, but GitHub makes it a great way to share your code too!



Sharing Your Code

To start sharing a repo you've created you need to publish it to GitHub.*

Some Git clients can publish directly, others require you to create the repo on the GitHub site.



* You also need to register for a GitHub account, but I'm not going to walk you through that. ;)



Search or jump to...

Pull requests Issues Marketplace Explore



MarkBennett / lets-git-this-party-started-demo

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Insights

Settings

A repository for demonstrating how to use Git and GitHub

Edit

Manage topics

6 commits

1 branch

0 releases

1 contributor

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



MarkBennett Add extra file

Latest commit f199c4d 11 hours ago

.gitattributes

Initial commit

2 days ago

LICENSE

Initial commit

2 days ago

README.md

Add resources

11 hours ago

dog.png

Add a picture of Toby

20 hours ago

extra.md

Add extra file

11 hours ago

README.md



Let's Git This Party Started

This repository contains contents and material I'll be using to demonstrate how to use *Git* and *Github*.

Getting Help

markbennett.ca



normal

Sharing Your Code



Public

Anyone can see this repository. You choose who can commit.



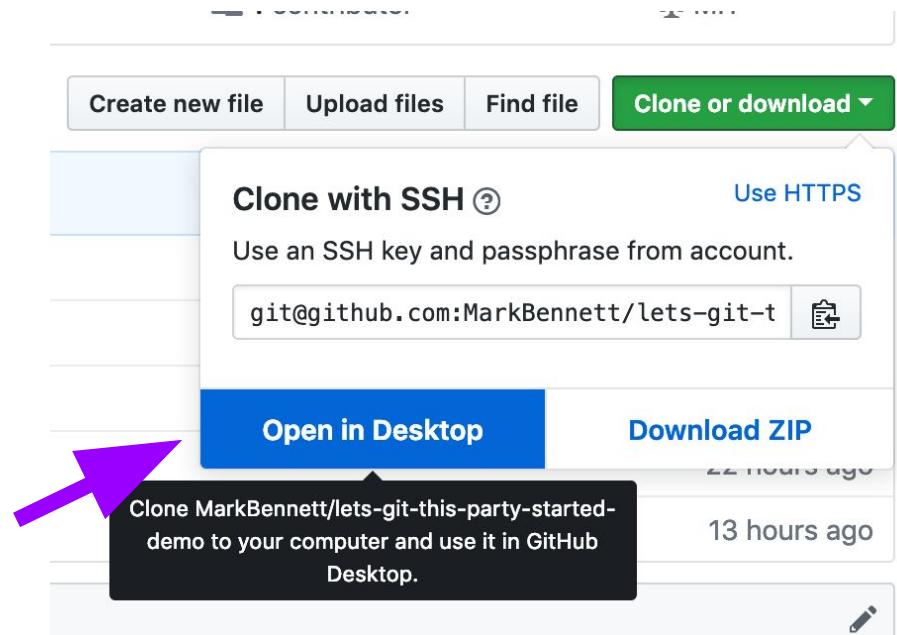
Private

You choose who can see and commit to this repository.

Sharing Your Code

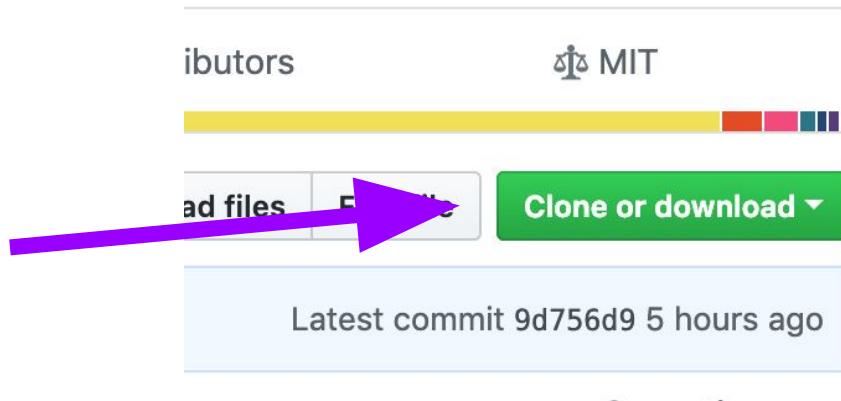
Now that it's up on GitHub your friends and co-workers can clone your repo to download your project on their computer.

There's even a shortcut to clone and open in GitHub Desktop!



Sharing Your Code

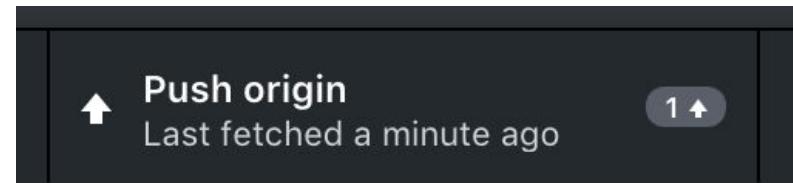
This works if you want to clone other peoples projects to use on your computer as well!



Sharing Your Code

As you work, you'll need to push commits and branches you make on your computer to GitHub.

Your work isn't shared until you push your changes.*



* This gets me *all the time*. :P

Your GitHub Repo

Your GitHub Repo

Every GitHub repository has a few interesting features on top of what Git provides.

- Issues
- Pull Requests
- Wiki
- Project webpage *

* Optional, and off by default



Search or jump to...

Pull requests Issues Marketplace Explore



MarkBennett / lets-git-this-party-started-demo

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Insights

Settings

A repository for demonstrating how to use Git and GitHub

Manage topics

6 commits

1 branch

0 releases

1 contributor

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

MarkBennett Add extra file

Latest commit f199c4d 11 hours ago

.gitattributes

Initial commit

2 days ago

LICENSE

Initial commit

2 days ago

README.md

Add resources

11 hours ago

dog.png

Add a picture of Toby

20 hours ago

extra.md

Add extra file

11 hours ago

README.md



Let's Git This Party Started

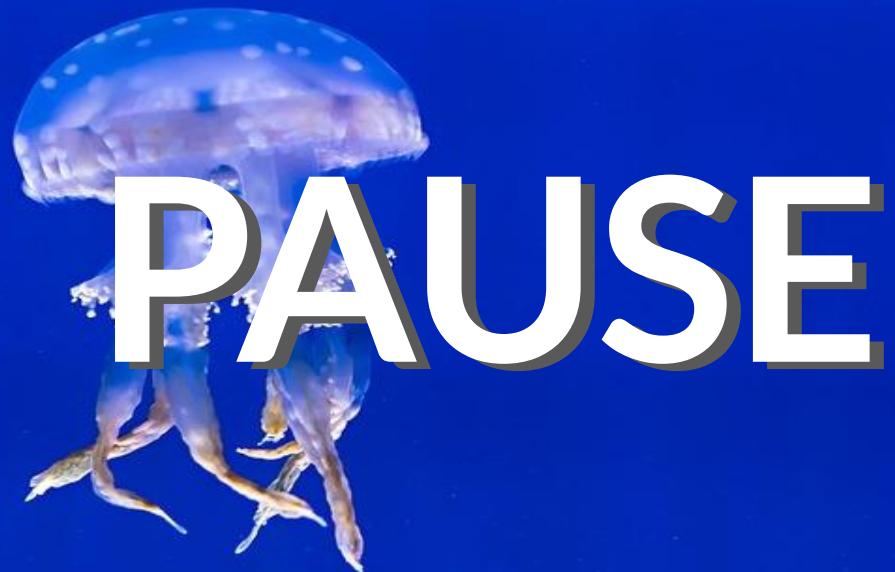
This repository contains contents and material I'll be using to demonstrate how to use *Git* and *GitHub*.

Getting Help

markbennett.ca



normal



Review

- GitHub hosts Git repositories and makes coding social.
- Repositories on GitHub have code as well as issues, pull requests, and documentation.
- You can quickly make your own repository to share your projects.
- Your repositories can be public or private.
- Clone a repository to your computer.
- **Don't forget to push changes!**

Questions?

Issues



Search or jump to...

Pull requests Issues Marketplace Explore



MarkBennett / lets-git-this-party-started-demo

Unwatch 1 Star 0 Fork 0

Code

Issues 2

Pull requests 1

Actions

Projects 0

Wiki

Insights

Settings

Find

is:issue is:open

Labels 8

Milestones 0

New issue

2 Open 0 Closed

Author ▾ Projects ▾ Labels ▾ Milestones ▾ Assignee ▾ Sort ▾

Another issue

#2 opened 4 minutes ago by MarkBennett

One issue

#1 opened 5 minutes ago by MarkBennett

ProTip! Type g i on any issue or pull request to go back to the issue listing page.

© 2019 GitHub, Inc. Terms Privacy Security Status Help



Contact GitHub Pricing API Training Blog About

markbennett.ca

normal

Issues

Issues... allow you to report and track issues!

You can apply labels to keep them organized, assign them to people, and add them to a milestone to do at a certain time.

Issues

Each issue includes discussion, and can be closed when it's addressed.

Note Labels, Assignees, etc. on the right.

One issue #1

Open MarkBennett opened this issue 19 minutes ago · 1 comment

MarkBennett commented 19 minutes ago · edited

This is the first issue I created!

Please keep on making awesome issues!!! 🎉

MarkBennett commented 19 seconds ago

Is this related to #2?

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Close issue Comment

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

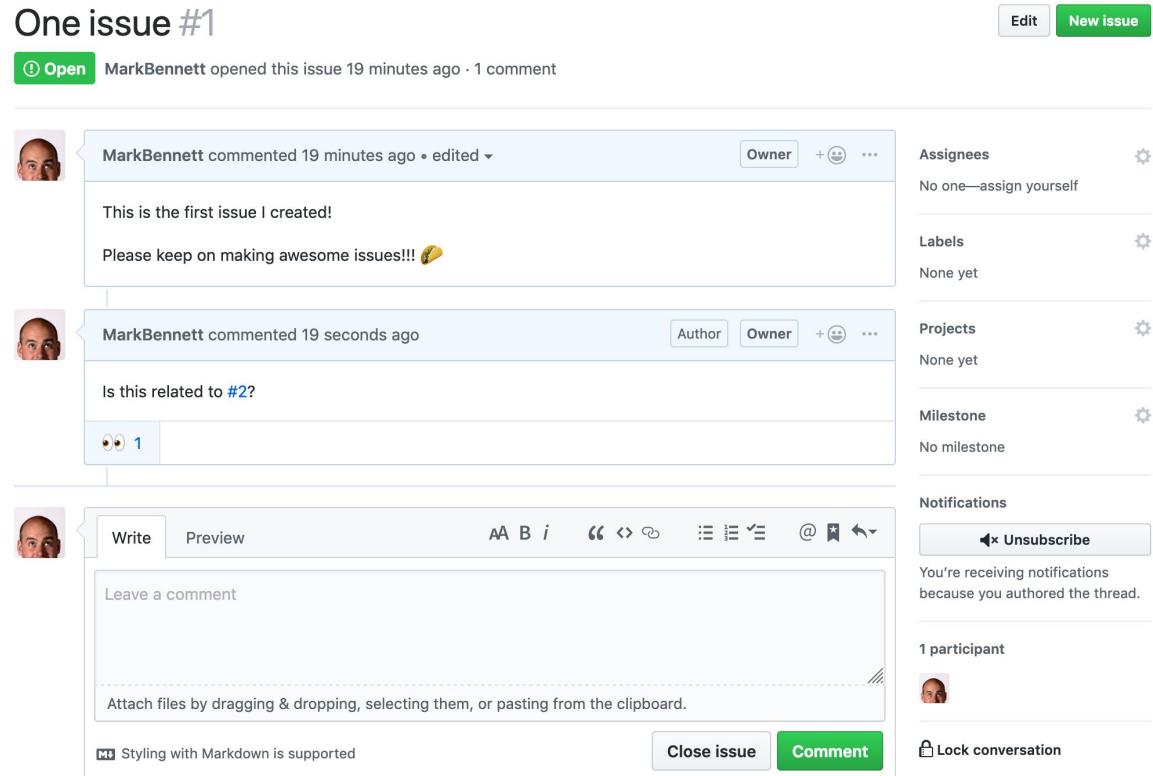
Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

Lock conversation



Issues

Issue numbers in any description or comment automatically link to the issue.

One issue #1

Open MarkBennett opened this issue 19 minutes ago · 1 comment

MarkBennett commented 19 minutes ago · edited

This is the first issue I created!

Please keep on making awesome issues!!! 🎉

MarkBennett commented 19 seconds ago

Is this related to #2?

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Edit New issue

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

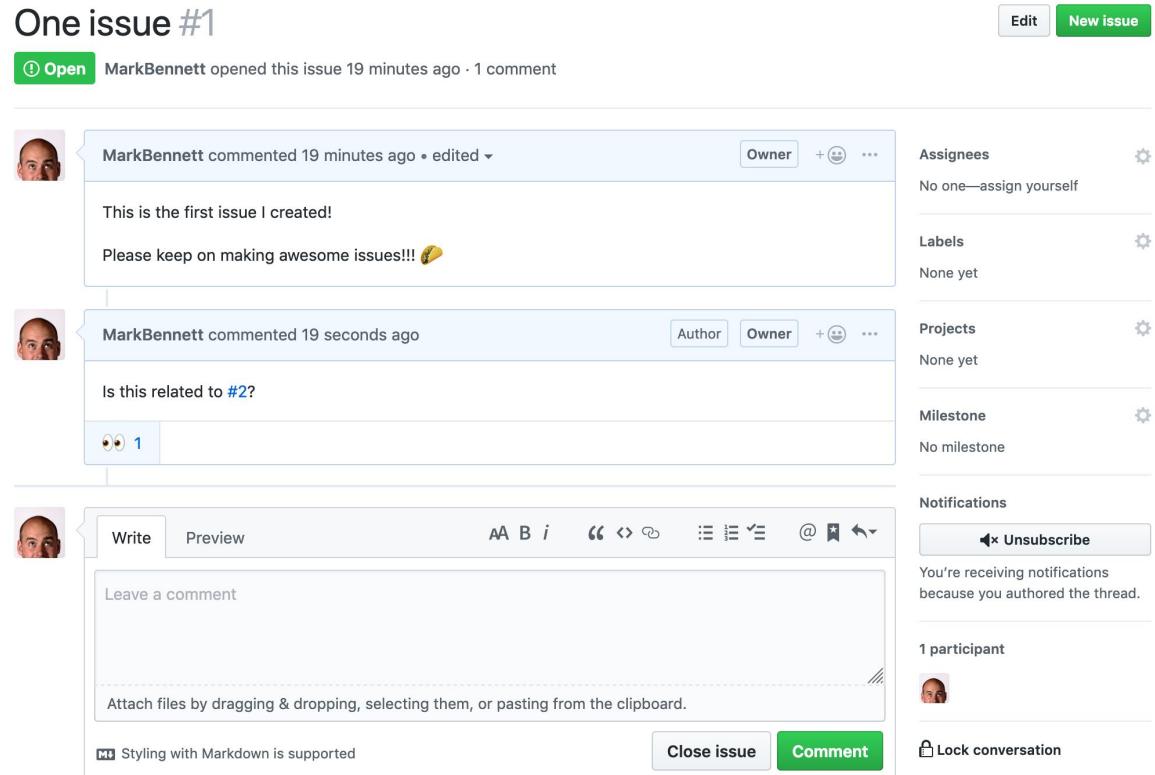
Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

Lock conversation



Issues

Use the “@” symbol
to mention someone.



Write Preview AA B i

What do you think of this @Mark

MarkBennett Mark Bennett



PAUSE

Review

- Use issues to organize your work.
- You can discuss issues.
- Mention other issues using a “#” and other people using “@”.
- Issues can be assigned to people and milestones.
- Close an issue when it’s resolved.

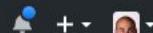
Questions?

Pull Requests



Search or jump to...

Pull requests Issues Marketplace Explore



MarkBennett / lets-git-this-party-started-demo

Unwatch 1

Star 0

Fork 0

Code

Issues 2

Pull requests 1

Actions

Projects 0

Wiki

Insights

Settings

Filters

is:pr is:open

Labels 8

Milestones 0

New pull request

1 Open 0 Closed

Author ▾

Projects ▾

Labels ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾

Make `extra.md` EXTRA special!

#3 opened a minute ago by MarkBennett

💡 ProTip! Filter pull requests by the default branch with [base:master](#)

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

markbennett.ca



normal

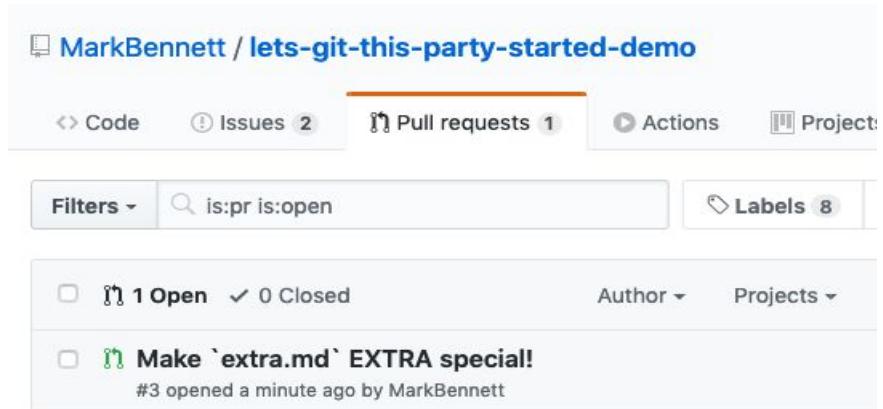
Pull Requests change everything!

markbennett.ca



Pull Requests

Pull requests are a request to merge one branch into another.



The screenshot shows a GitHub repository named "MarkBennett / lets-git-this-party-started-demo". The repository has 2 issues and 1 pull request. The pull request is titled "Make `extra.md` EXTRA special!" and was opened by MarkBennett a minute ago. The repository also has 8 labels. A search bar at the top shows the filter "is:pr is:open". A pro tip at the bottom suggests filtering pull requests by default.

MarkBennett / lets-git-this-party-started-demo

Code Issues 2 Pull requests 1 Actions Projects

Filters is:pr is:open Labels 8

1 Open 0 Closed Author Projects

Make `extra.md` EXTRA special!
#3 opened a minute ago by MarkBennett

💡 ProTip! Filter pull requests by the default

Pull Requests

You can discuss and link to requests like a normal issue.

When you're done you can merge or close the request.

The screenshot shows a GitHub pull request page for a repository named "lets-git-this-party-started-demo". The pull request is titled "Make `extra.md` EXTRA special! #3" and is marked as "Open". It has 1 commit from "MarkBennett" and 1 file changed ("Update extra.md"). A comment from "MarkBennett" is visible, stating "No description provided." Below the comment, there's a note about continuous integration: "Continuous integration has not been set up. Several apps are available to automatically catch bugs and enforce style." A green checkmark indicates "This branch has no conflicts with the base branch" and "Merging can be performed automatically." At the bottom of the pull request view, there are buttons for "Merge pull request" and "Close pull request". The bottom half of the screenshot shows the GitHub rich text editor interface with "Write" and "Preview" tabs, a comment input field, and a file attachment section.

Pull Requests

GitHub shows the changes included in the branch.

These changes update as the source branch is changed.

A screenshot of a GitHub pull request interface. At the top, a green button says "Open" and the title is "Make `extra.md` EXTRA special! #3". A pink arrow points from the top right towards the "Files changed" tab. Below the title, there are tabs for "Conversation" (1), "Commits" (1), "Checks" (0), and "Files changed" (1). The "Files changed" tab is selected. The main area shows a diff for a file named "extra.md". The diff highlights changes in a green-to-red color scheme. The first change is a new file being added: "1 + # Extra! Extra!". A comment from user "MarkBennett" is shown below the diff, saying "Great idea. This is extra special!". There is a reply input field and a "Resolve conversation" button. At the bottom, a pro tip says "ProTip! Use [n] and [p] to navigate between commits in a pull request."

Pull Requests

You can comment on any line.

Comments can be a single comment
or you can start a review.



Make `extra.md` EXTRA special! #3

[Open](#) MarkBennett wants to merge 1 commit into master from adding-extra-info

Conversation 1 Commits 1 Checks 0 Files changed 1

Changes from all commits ▾ File filter... ▾ Jump to... +3 -1

Diff settings Review changes

4 extra.md Show comments

```
... ... @@ -1 +1,3 @@
- This is an extra file for later.
+ # Extra! Extra!
```

MarkBennett 10 hours ago Author Owner
Great idea. This is extra special!

Reply...

Resolve conversation

Start a new conversation

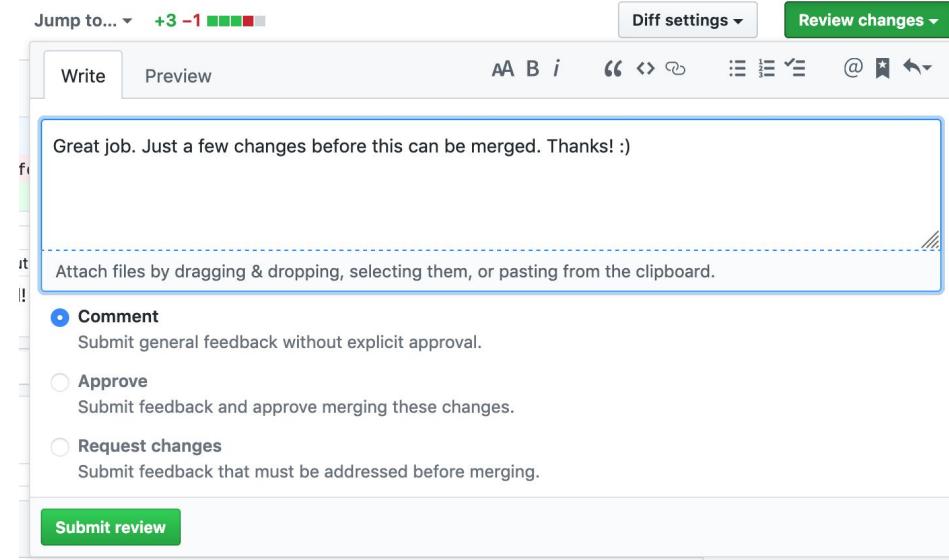
```
2 +
3 + This is an extra file for later. It's *EXTRA* special!
```

ProTip! Use [n](#) and [p](#) to navigate between commits in a pull request.

Pull Requests

Reviews include many comments.

At the end of review you can approve the pull request, require changes, or just comment.



Pull Requests

Reviews appear in the pull request discussion.

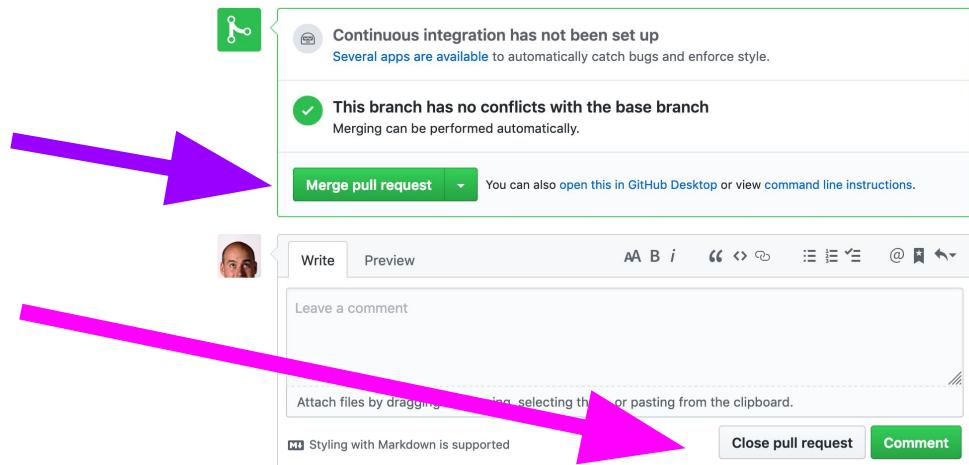
This makes it easy to see if a pull request has been reviewed and approved or if changes are required.

A screenshot of a GitHub pull request titled "Make `extra.md` EXTRA special! #3". The pull request is open, showing 1 commit from MarkBennett merged into the master branch from the "adding-extra-info" repository. The conversation tab is selected, showing a comment from MarkBennett: "Just some changes to make sure this file is as special as it needs to be! 🎉". Below the comment, there is a commit titled "Update extra.md" by MarkBennett, which has been reviewed just now. A pink arrow points to the review section of the commit, where MarkBennett says "Great idea. This is extra special!". There is also a "Resolve conversation" button at the bottom of the review section.

Pull Requests

When a pull request is done the repository you can merge the pull request.

You can also close the pull request if you're not going to use the work.



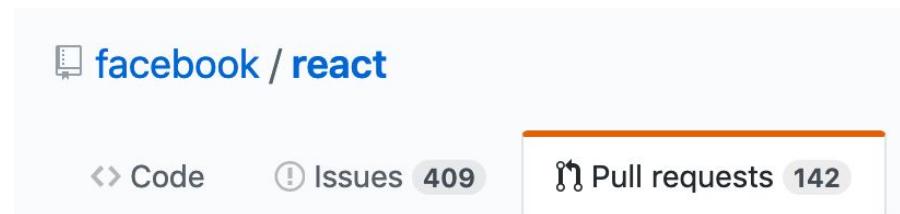


But there's more!

Pull Requests

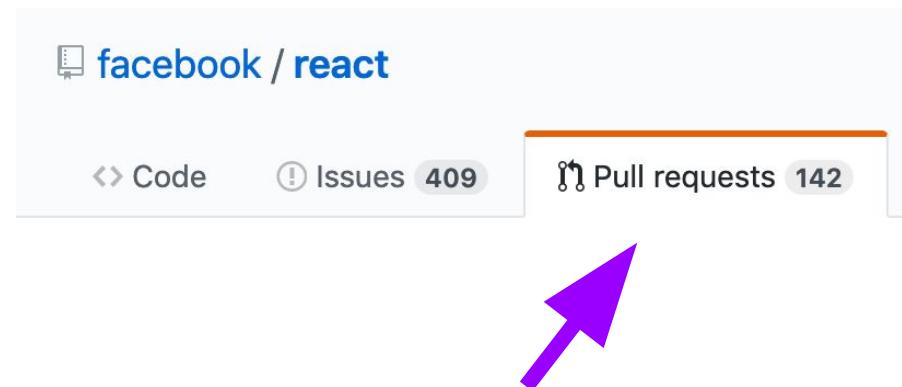
If your repository is public, then people can help you out and submit pull requests!

You're users submit pull requests, you choose which to merge and close.



Pull Requests

This is why open source projects like GitHub so much. It's so easy to accept and review work from the public! 😊



Pull Requests

Open Source on GitHub

=

Free Bug Fixes & Features

Pull Requests

Teams use pull requests to review and share their work.

You can even make a draft pull request to share and get feedback while you're still working.

asting from the clipboard.

Create pull request

✓ Create pull request

Open a pull request that is ready for review

Create draft pull request

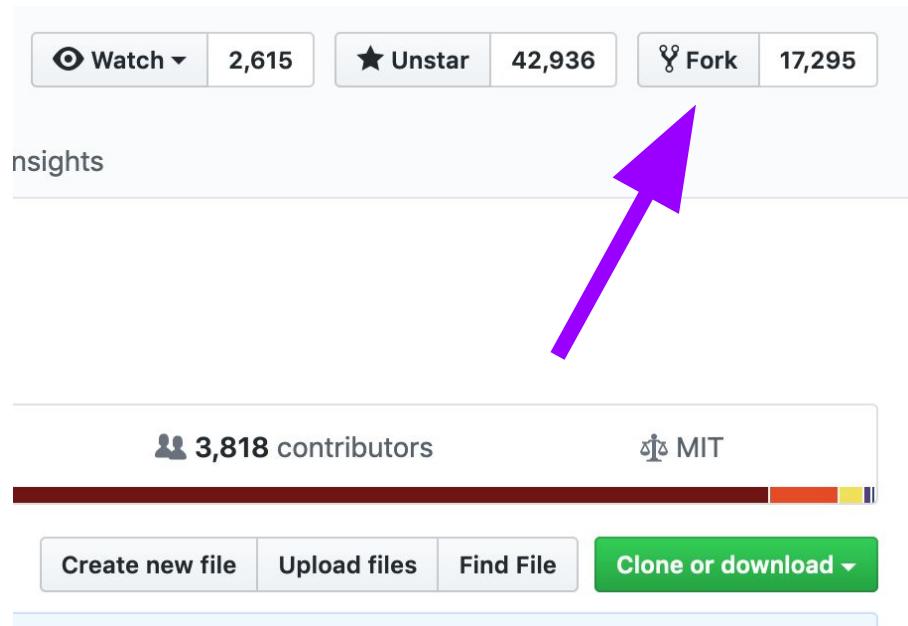
Cannot be merged until marked ready for review

ile.md

Forking A Repository

Start contributing by forking a repository. This makes your own copy of the repo you can commit to.

Commit your changes to your repo, then you can create a pull request to the repo you forked.



A wide-angle photograph of a desert landscape. The foreground is filled with the undulating patterns of sand dunes. In the middle ground, a path or ridge leads towards a range of hills or dunes under a vast sky. The sky is filled with large, billowing clouds, ranging from white to dark grey, set against a bright blue background.

PAUSE

Review

- Pull requests define GitHub for many people.
- A pull request is a request to merge a branch into another.
- Teams use pull requests to review work.
- A pull request can carry on for a while well work is done.
- Open source projects use pull requests to receive contributions.

Questions?

Exploring GitHub

Exploring GitHub

Notifications alert you to important information and changes in the repositories you own and watch on GitHub.

Notifications Watching

Unread	16
Read	2
Participating	3
Saved for later	1
All notifications	

MarkBennett/markbennett.github.com

Potential security vulnerability found in the key

react-native-community/react-native-releases

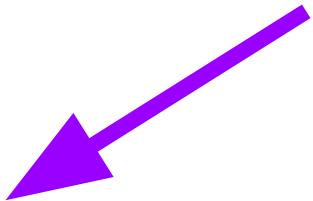
0.59.x Discussion

edmontongo/presentations

January 2019

MarkBennett/rumbl...	2
yegrb/leduccountryl...	2
MarkBennett/yegtra...	1

Exploring GitHub



The screenshot shows the GitHub homepage with a dark theme. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the header, there are three main featured sections:

- HashiCorp ships Vault 1.0**: An image of a circuit board with arrows pointing to specific components. Text: "RELEASE RADAR HashiCorp ships Vault 1.0. Securely save and access your app's secrets, like database passwords and API keys."
- She uses music and technology to tell stories**: A photo of a woman with braided hair wearing a hat, smiling. Text: "Meet Jamica El >"
- Style your site 8-bit**: An image of a game progress bar and a set of colorful CSS icons. Text: "NES.css' initial release offers a CSS framework with a classic gaming aesthetic."

Based on your interests

 [deltachat / deltachat-desktop](#)

 [reactjs / reactjs.org-translation](#)

 [pinterest / gestalt](#)
A set of React UI components that

 [Polymer / lit-element](#)
A simple base class for creating

 [terser-js / terser](#)
JavaScript parser, m

Exploring GitHub

Topics allow you to find new repositories and projects.

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

The screenshot shows the GitHub Topics page. At the top, there's a navigation bar with a search bar, pull requests, issues, marketplace, and explore links. Below the header, the title "Topics" is displayed, followed by the subtext "Browse the top used topics on GitHub". There are three main cards for featured topics:

- OpenGL**: OpenGL is a software interface that allows a programmer to communicate with graphics hardware.
- Jekyll**: Jekyll is a simple, blog-aware static site generator.
- Python**: Python is a dynamically typed programming language.

Below these, there's a section titled "All featured topics" with cards for "3D" and "Ajax". To the right, there's a "Popular topics" sidebar with tags for nodejs, javascript, python, java, docker, golang, typescript, html5, css3, and paket.

A wide-angle photograph of a desert landscape. The foreground is filled with the undulating patterns of sand dunes. In the middle ground, a path or ridge leads towards a range of hills or dunes under a vast sky. The sky is filled with large, billowing clouds, with patches of blue visible between them.

PAUSE

Review

- Branches let you work and commit independently from one another.
- Branches are used for bug fixes, experiments, new features, work in progress.
- New commits are added to the branch you've checked out.
- You always have one branch checked out.
- By default every repository has a “*master*” branch.

Questions?

Next Steps

Advanced Git

- .gitignore
- git log
- git stash
- git blame
- git bisect
- git rebase -i
- git commit --amend
- git push --force (don't!)
- git remote add upstream

Advanced GitHub

- GitHub pages
- .gitignore
- Forking repositories
- CONTRIBUTING
- LICENSE
- .github

Getting Help

- Dev Edmonton Slack
- Learn Git
- Learn GitHub

Resources

- [Atlassian Git Tutorials](#)
- [Git entry on Wikipedia](#)
- [Learn Git Branching](#)
- [github/gitignore](#)
- [Oh, s**t, git!](#)
- [Zero To Git](#)

Acknowledgements

- Linus Torvald
- Dev Edmonton Slack

Get Help

devedmonton.com

A community of hundreds of local
developers and designers.
Including EDMUG!

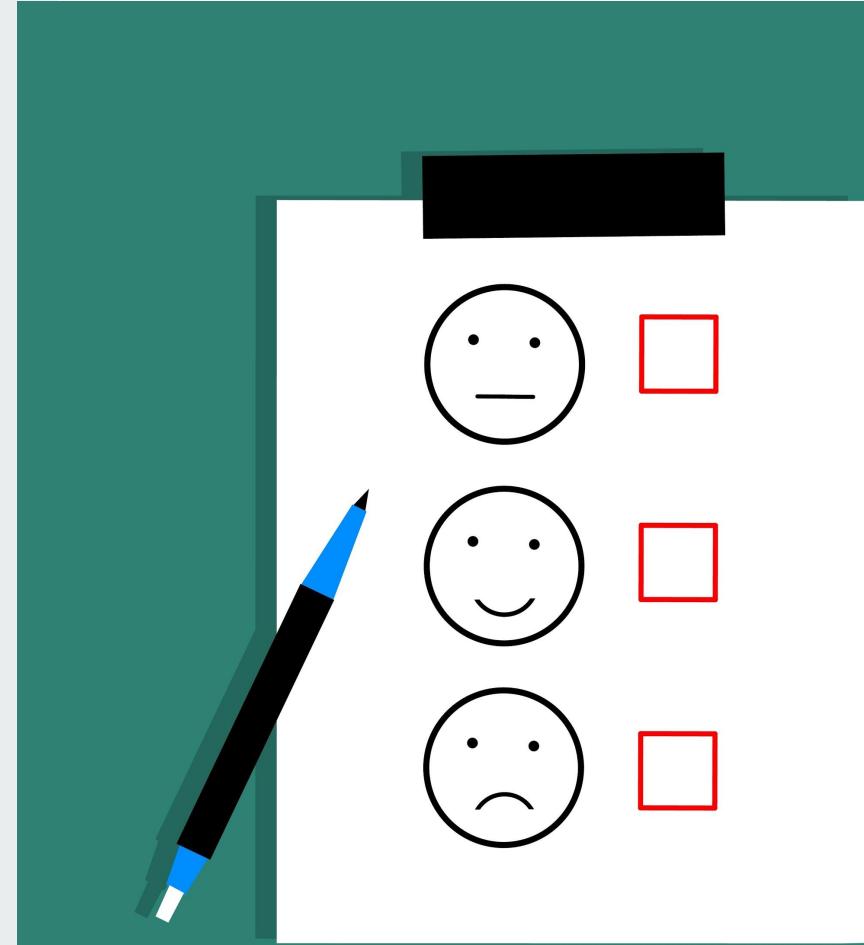
Slack link is at the bottom.



Feedback

Git and GitHub are challenging topics to teach.

@MarkBennett or DES Slack



Thank-you!

Questions?

Is Git secure?

Yes.

**Each commit is digitally
fingerprinted, and repositories
are transferred securely.**
