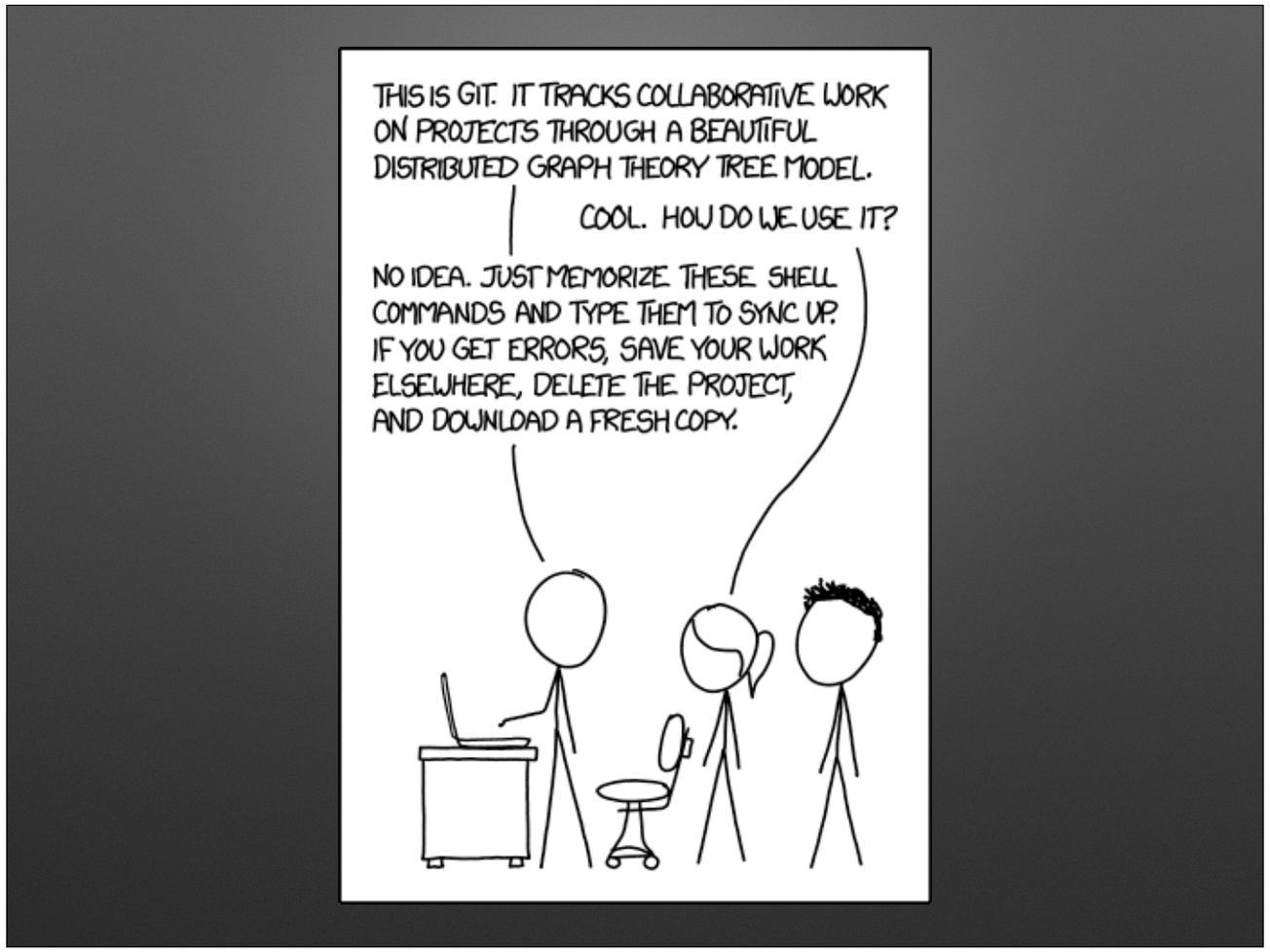
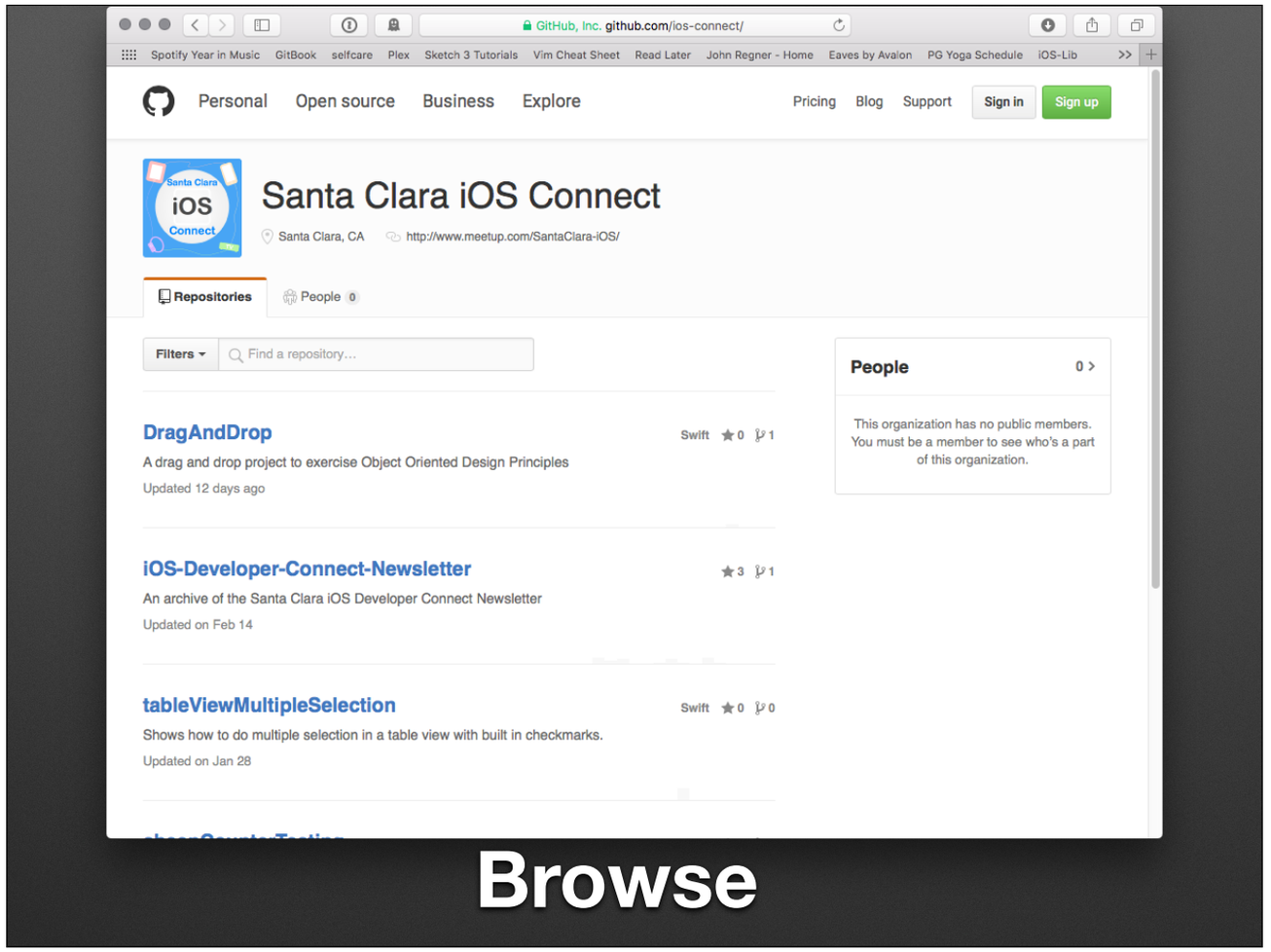


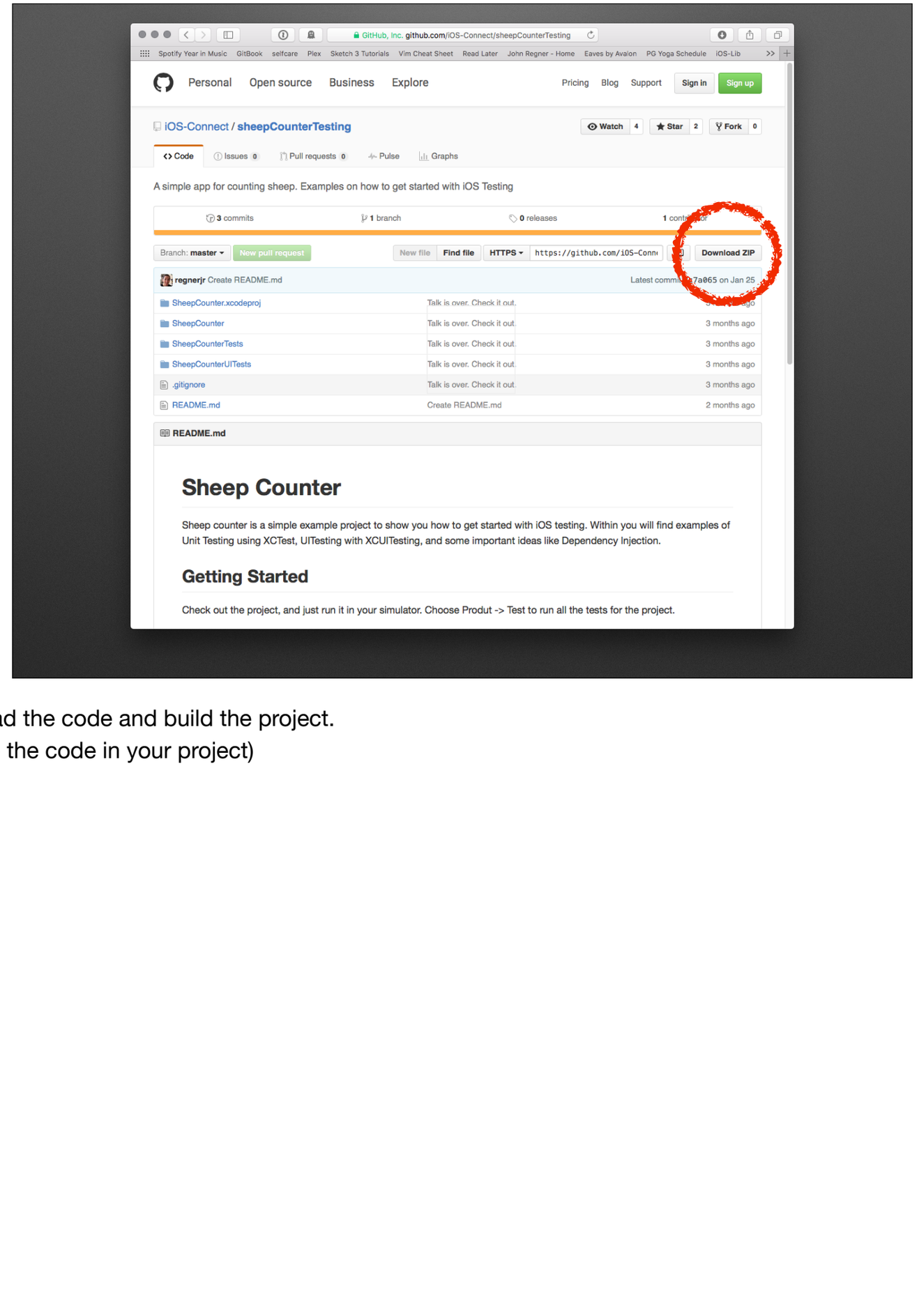
Intro slide for humor.

We have Git and GitHub (this is an OctoCat)

Git is a tool, that lives on your computer, like word, or ruby.
GitHub is a website.

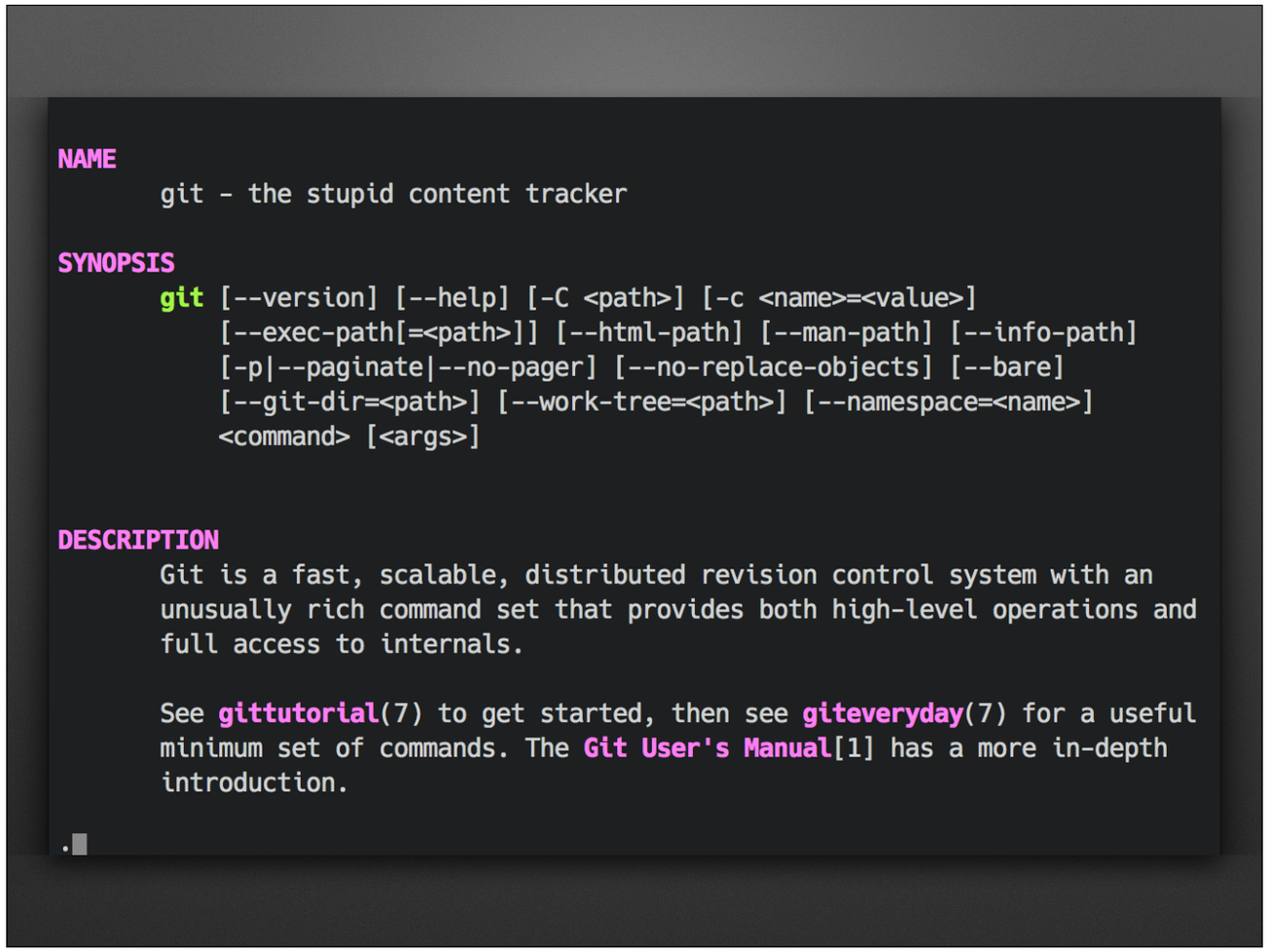


Browse code repositories on github



Grab a copy of any project. You can read the code and build the project.
(Depending on the license, you can use the code in your project)

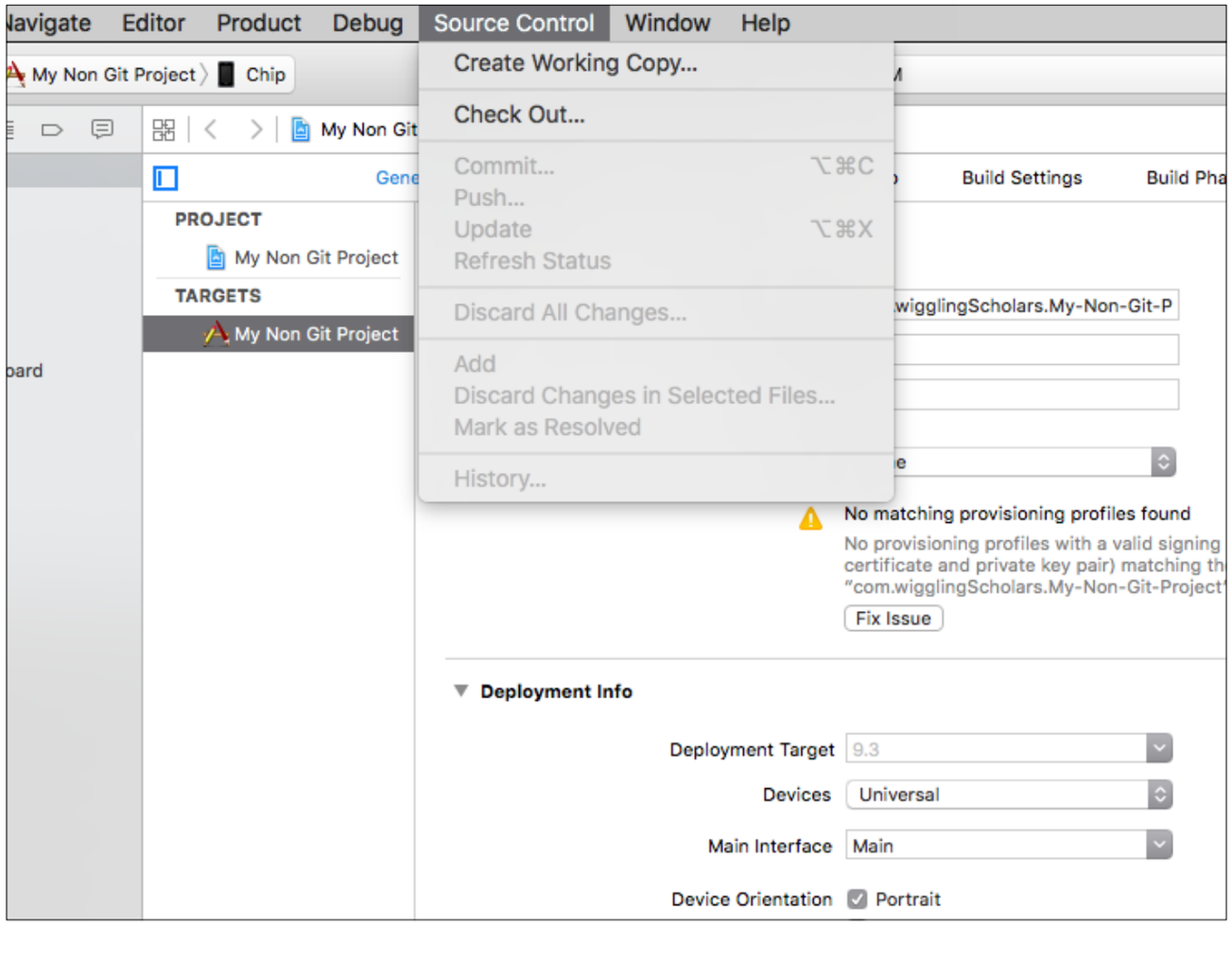
Getting Started



This is the git user manual.
“stupid content tracker”

git init

Run `git init` in any folder to tell git to track it


Navigate
Editor
Product
Debug
Source Control
Window
Help
Create Working Copy...
Check Out...
Commit...
⌥⌘C
Push...
Update
⌥⌘X
Refresh Status
Discard All Changes...
Add
Discard Changes in Selected Files...
Mark as Resolved
History...
My Non Git Project
Chip
PROJECT
My Non Git Project
TARGETS
My Non Git Project
Build Settings
Build Pha
No matching provisioning profiles found
No provisioning profiles with a valid signing
certificate and private key pair) matching th
"com.wigglingScholars.My-Non-Git-Project"
Fix Issue
Deployment Info
Deployment Target
9.3
Devices
Universal
Main Interface
Main
Device Orientation
Portrait

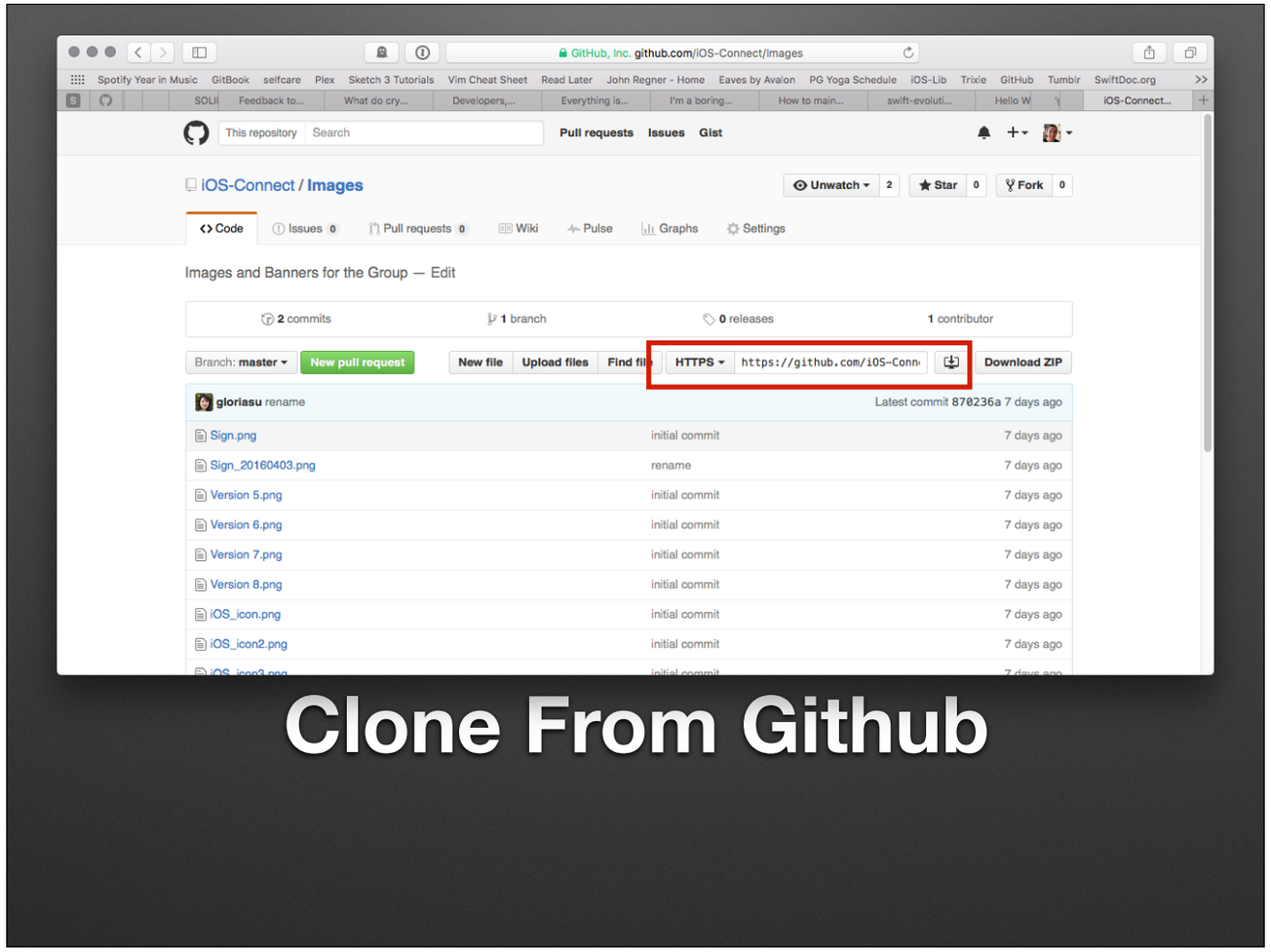
Existing Git Repos

- Fork from GitHub (talk about this later)
- Clone from anywhere, usually a URL
  http://someURL.com/projects/linux.git

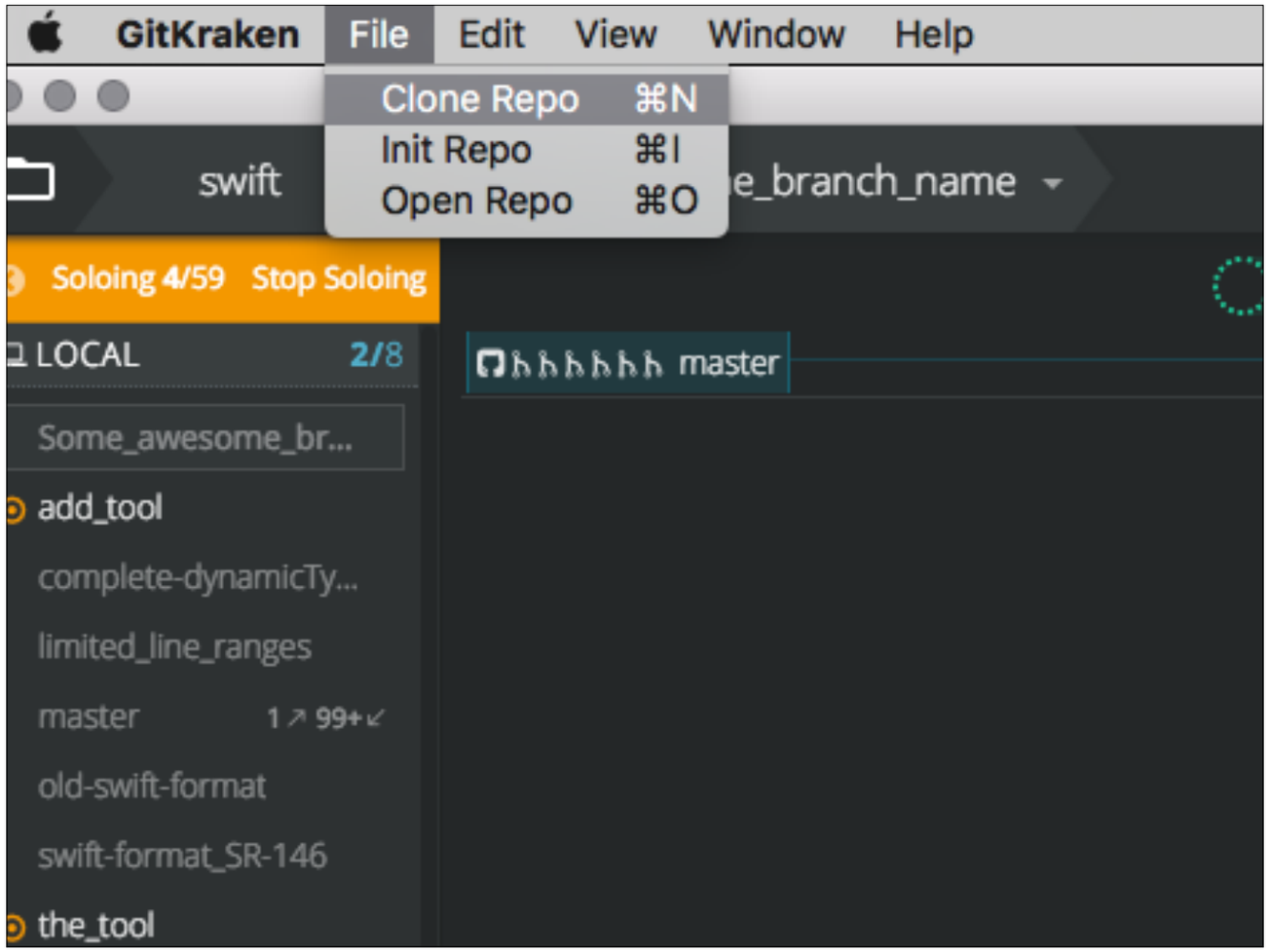
git clone <someURL>

Run `git clone` where you want the project to live

makes you a copy



Clone From Github


GitKraken
File
Edit
View
Window
Help
Clone Repo ⌘N
Init Repo ⌘I
Open Repo ⌘O
swift
e_branch_name
Soloing 4/59 Stop Soloing
LOCAL 2/8
master
Some_awesome_br...
add_tool
complete-dynamicTy...
limited_line_ranges
master 1↗ 99+↙
old-swift-format
swift-format_SR-146
the_tool

git clone <SomeURL>

Make a full copy of the project on your computer

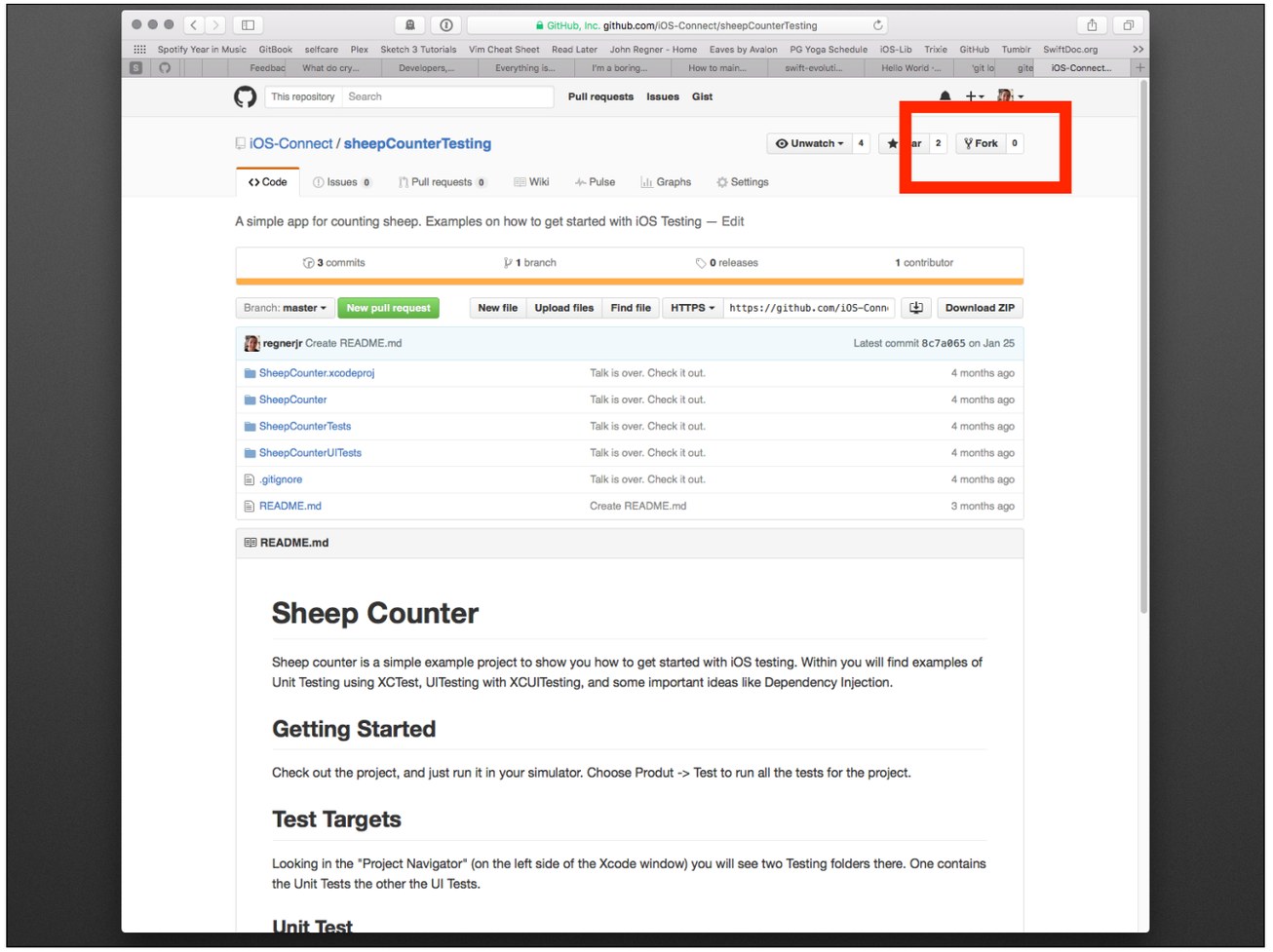
- an example project
- a Library you want to use
- a friend's code to help debug

Since this is a full git copy,
you can continue working just the way
you normally would

git fork

A server side copy on github.

not a terminal command (this slide is a lie)


GitHub, Inc. github.com/iOS-Connect/sheepCounterTesting
This repository
Search
Pull requests
Issues
Gist
iOS-Connect / sheepCounterTesting
Unwatch 4
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
A simple app for counting sheep. Examples on how to get started with iOS Testing — Edit
3 commits
1 branch
0 releases
1 contributor
Branch: master
New pull request
New file
Upload files
Find file
HTTPS
https://github.com/iOS-Conn
Download ZIP
regnerjr Create README.md
Latest commit 8c7a065 on Jan 25
SheepCounter.xcodeproj Talk is over. Check it out. 4 months ago
SheepCounter Talk is over. Check it out. 4 months ago
SheepCounterTests Talk is over. Check it out. 4 months ago
SheepCounterUITests Talk is over. Check it out. 4 months ago
.gitignore Talk is over. Check it out. 4 months ago
README.md Create README.md 3 months ago
README.md
Sheep Counter
Sheep counter is a simple example project to show you how to get started with iOS testing. Within you will find examples of Unit Testing using XCTest, UITesting with XCUITesting, and some important ideas like Dependency Injection.
Getting Started
Check out the project, and just run it in your simulator. Choose Produt -> Test to run all the tests for the project.
Test Targets
Looking in the "Project Navigator" (on the left side of the Xcode window) you will see two Testing folders there. One contains the Unit Tests the other the UI Tests.
Unit Test

Review - Setup

- `git init` a local project folder
- `git fork` to make a copy on github
- `git clone <someURL>` to make a copy from somewhere to your local machine

Sweet Success

Sweet Now you have a git repo

What do we do with it?

Everyday Git

More like every hour.

All about the Commits

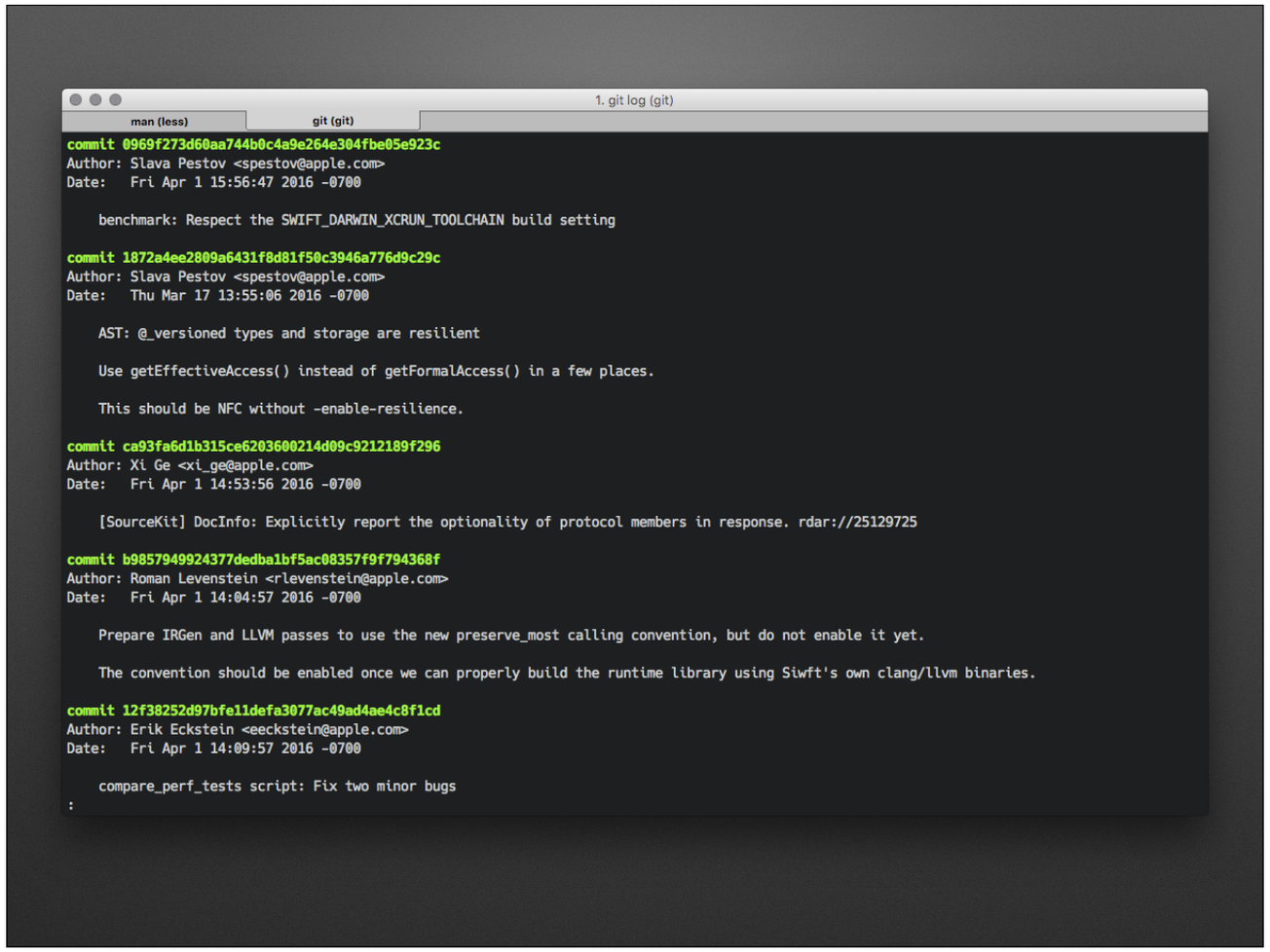
Write new code or fix bugs. Then you will want to "make a snapshot"
git calls this a commit.
These commits are the main pieces that git works with.

A Commit Object has

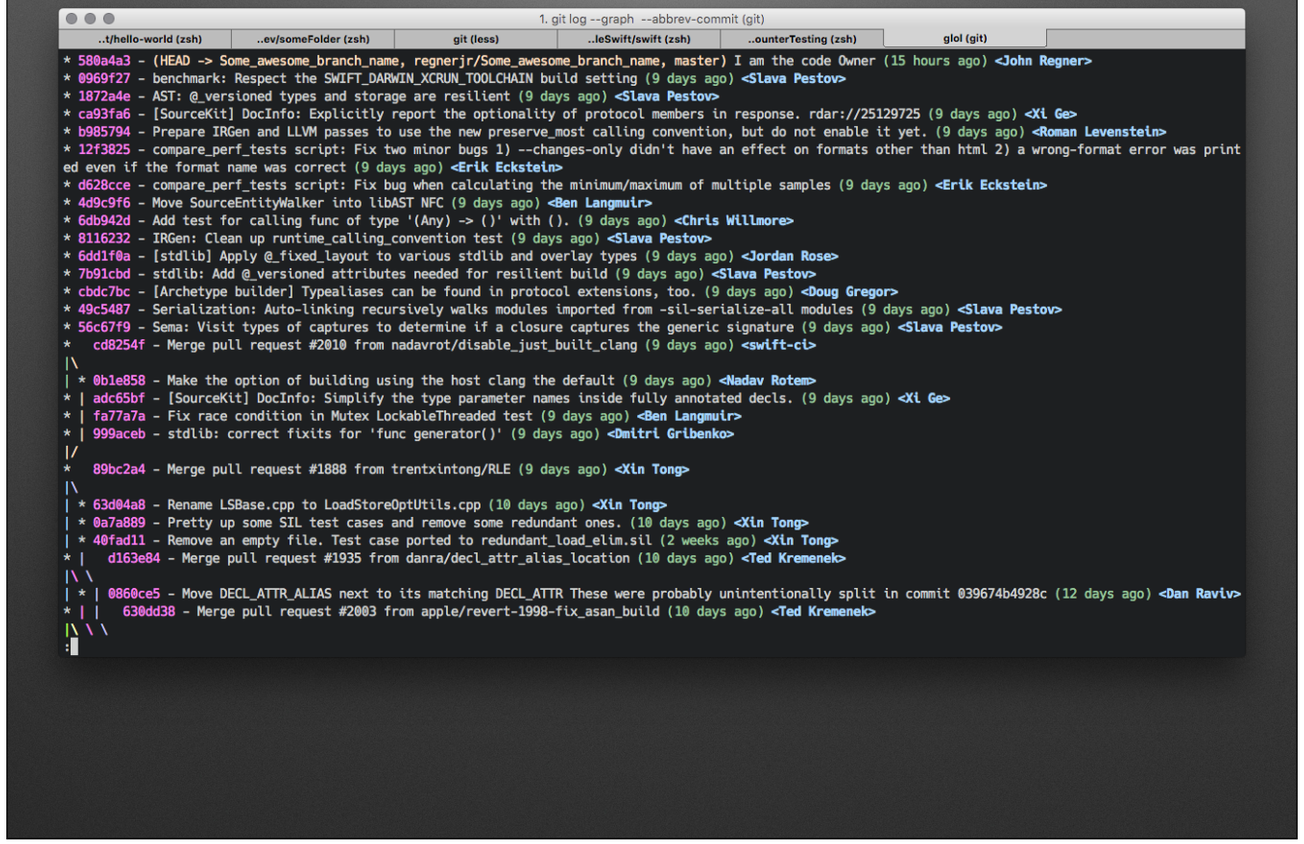
- An Author - Git will ask you to configure this
- A complete project
- A unique hash
- A Message

* An author - Git will ask you to set these up if they are not yet configured.
* A complete project - The full source is here (No computed diffs)
* A unique hash
* A message

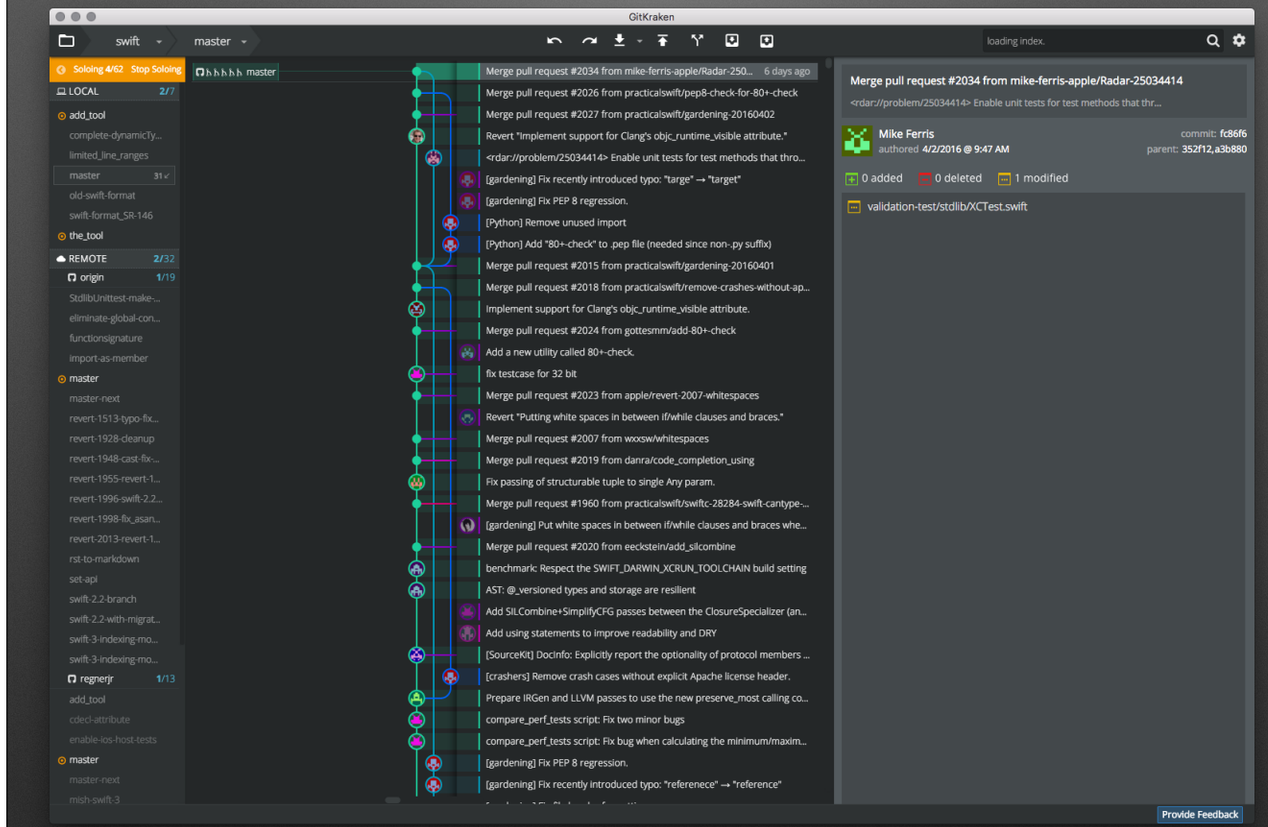
You probably want a Tool

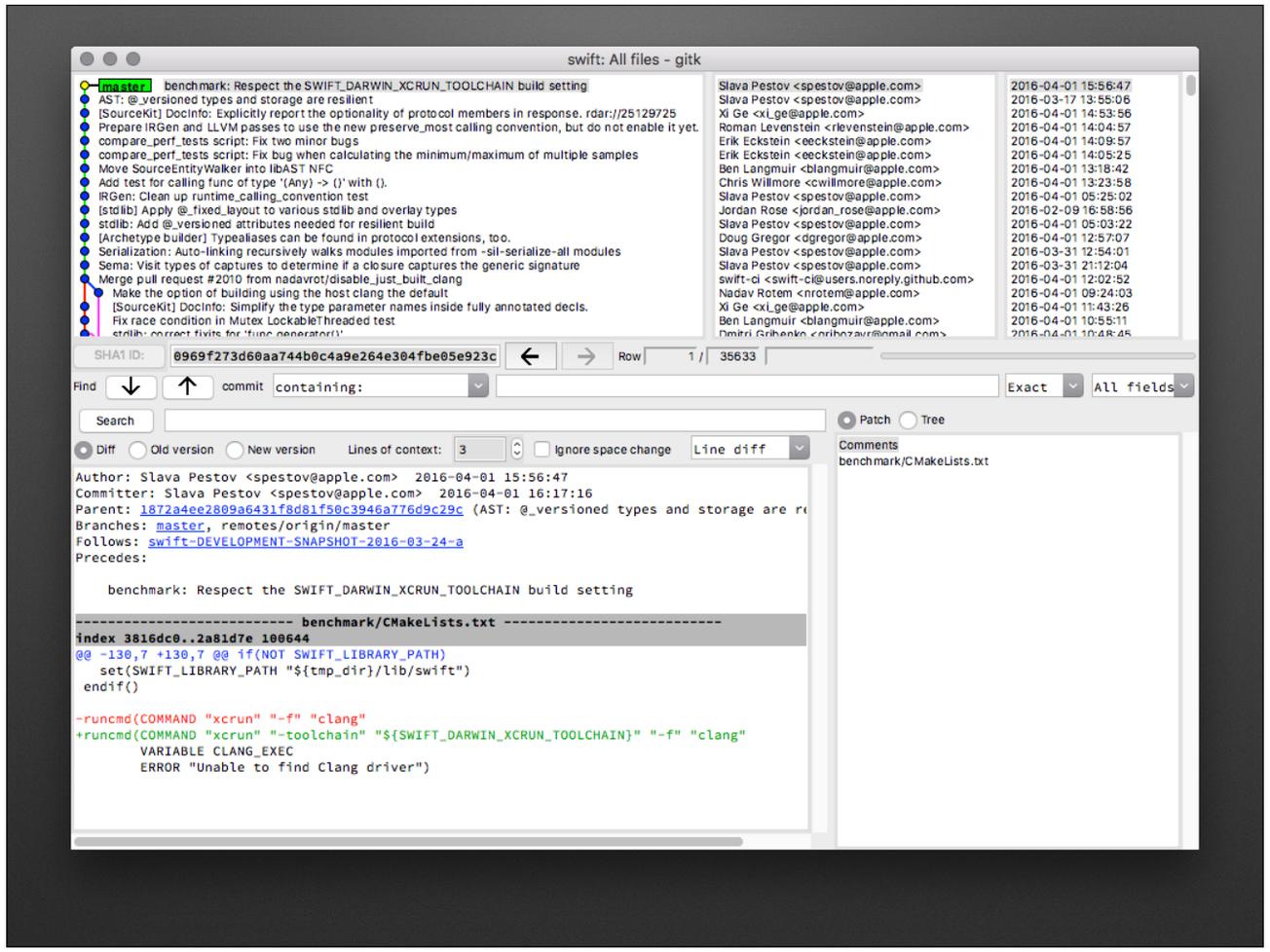


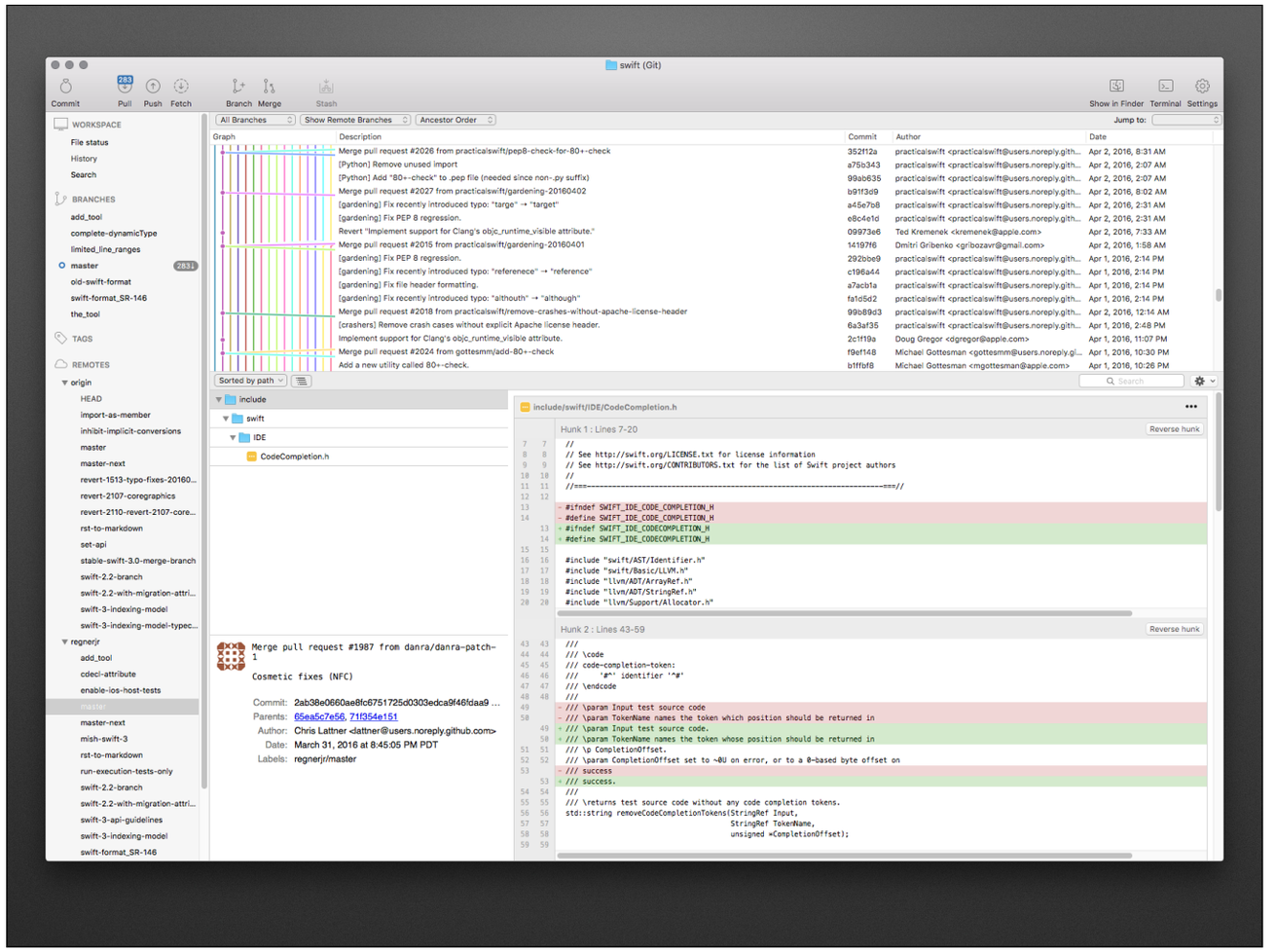
git log is not super good for this. There are a ton of commands to change the way this looks.



Use the right flags we can shorten this a bit.


GitKraken
swift
master
loading index.
Soloing 4/62 Stop Soloing
LOCAL 2/7
add_tool
complete-dynamicFly...
limited_line_ranges
master 31
old-swift-format
swift-format_SR-146
the_tool
REMOTE 2/32
origin 1/19
StdlibUnittest-make-...
eliminate-global-con...
functionsignature
import-as-member
master
master-next
revert-1513-typo-fix...
revert-1928-cleanup
revert-1948-cast-fix-...
revert-1955-revert-1...
revert-1996-swift-2.2...
revert-1998-fix_asan...
revert-2013-revert-1...
rst-to-markdown
set-api
swift-2.2-branch
swift-2.2-with-migrat...
swift-3-indexing-mo...
swift-3-indexing-mo...
regnerjr 1/13
add_tool
cdecl-attribute
enable-ios-host-tests
master
master-next
mish-swift-3
master 6 days ago
Merge pull request #2034 from mike-ferris-apple/Radar-250...
Merge pull request #2026 from practicalswift/pep8-check-for-80+-check
Merge pull request #2027 from practicalswift/gardening-20160402
Revert "Implement support for Clang's objc_runtime_visible attribute."
<rdar://problem/25034414> Enable unit tests for test methods that thro...
[gardening] Fix recently introduced typo: "targe" → "target"
[gardening] Fix PEP 8 regression.
[Python] Remove unused import
[Python] Add "80+-check" to .pep file (needed since non-.py suffix)
Merge pull request #2015 from practicalswift/gardening-20160401
Merge pull request #2018 from practicalswift/remove-crashes-without-ap...
Implement support for Clang's objc_runtime_visible attribute.
Merge pull request #2024 from gottesmm/add-80+-check
Add a new utility called 80+-check.
fix testcase for 32 bit
Merge pull request #2023 from apple/revert-2007-whitespaces
Revert "Putting white spaces in between if/while clauses and braces."
Merge pull request #2007 from woosw/whitespaces
Merge pull request #2019 from danra/code_completion_using
Fix passing of structurable tuple to single Any param.
Merge pull request #1960 from practicalswift/swiftc-28284-swift-cantype-...
[gardening] Put white spaces in between if/while clauses and braces whe...
Merge pull request #2020 from eckstein/add_silcombine
benchmark: Respect the SWIFT_DARWIN_XCRUN_TOOLCHAIN build setting
AST: @_versioned types and storage are resilient
Add SILCombine+SimplifyCFG passes between the ClosureSpecializer (an...
Add using statements to improve readability and DRY
[SourceKit] DocInfo: Explicitly report the optionality of protocol members ...
[crashers] Remove crash cases without explicit Apache license header.
Prepare IRGen and LLVM passes to use the new preserve_most calling co...
compare_perf_tests script: Fix two minor bugs
compare_perf_tests script: Fix bug when calculating the minimum/maxim...
[gardening] Fix PEP 8 regression.
[gardening] Fix recently introduced typo: "referenece" → "reference"
Merge pull request #2034 from mike-ferris-apple/Radar-25034414
<rdar://problem/25034414> Enable unit tests for test methods that thr...
Mike Ferris
authored 4/2/2016 @ 9:47 AM
commit: fc86f6
parent: 352f12, a3b880
0 added
0 deleted
1 modified
validation-test/stdlib/XCTest.swift
Provide Feedback


swift: All files - gitk
master
benchmark: Respect the SWIFT_DARWIN_XCRUN_TOOLCHAIN build setting
AST: @_versioned types and storage are resilient
[SourceKit] DocInfo: Explicitly report the optionality of protocol members in response. rdar://25129725
Prepare IRGen and LLVM passes to use the new preserve_most calling convention, but do not enable it yet.
compare_perf_tests script: Fix two minor bugs
compare_perf_tests script: Fix bug when calculating the minimum/maximum of multiple samples
Move SourceEntityWalker into libAST NFC
Add test for calling func of type '(Any) -> ()' with ().
IRGen: Clean up runtime_calling_convention test
[stdlib] Apply @_fixed_layout to various stdlib and overlay types
stdlib: Add @_versioned attributes needed for resilient build
[Archetype builder] Typealiases can be found in protocol extensions, too.
Serialization: Auto-linking recursively walks modules imported from -sil-serialize-all modules
Sema: Visit types of captures to determine if a closure captures the generic signature
Merge pull request #2010 from nadavrot/disable_just_built_clang
Make the option of building using the host clang the default
[SourceKit] DocInfo: Simplify the type parameter names inside fully annotated decls.
Fix race condition in Mutex LockableThreaded test
Slava Pestov <spestov@apple.com> 2016-04-01 15:56:47
Slava Pestov <spestov@apple.com> 2016-03-17 13:55:06
Xi Ge <xi_ge@apple.com> 2016-04-01 14:53:56
Roman Levenstein <rlevenstein@apple.com> 2016-04-01 14:04:57
Erik Eckstein <eeckstein@apple.com> 2016-04-01 14:09:57
Erik Eckstein <eeckstein@apple.com> 2016-04-01 14:05:25
Ben Langmuir <blangmuir@apple.com> 2016-04-01 13:18:42
Chris Willmore <cwillmore@apple.com> 2016-04-01 13:23:58
Slava Pestov <spestov@apple.com> 2016-04-01 05:25:02
Jordan Rose <jordan_rose@apple.com> 2016-02-09 16:58:56
Slava Pestov <spestov@apple.com> 2016-04-01 05:03:22
Doug Gregor <dgregor@apple.com> 2016-04-01 12:57:07
Slava Pestov <spestov@apple.com> 2016-03-31 12:54:01
Slava Pestov <spestov@apple.com> 2016-03-31 21:12:04
swift-ci <swift-ci@users.noreply.github.com> 2016-04-01 12:02:52
Nadav Rotem <nrotem@apple.com> 2016-04-01 09:24:03
Xi Ge <xi_ge@apple.com> 2016-04-01 11:43:26
Ben Langmuir <blangmuir@apple.com> 2016-04-01 10:55:11
SHA1 ID: 0969f273d60aa744b0c4a9e264e304fbe05e923c
Row 1 / 35633
Find commit containing: Exact All fields
Search
Patch Tree
Diff Old version New version Lines of context: 3 Ignore space change Line diff
Comments
benchmark/CMakeLists.txt
Author: Slava Pestov <spestov@apple.com> 2016-04-01 15:56:47
Committer: Slava Pestov <spestov@apple.com> 2016-04-01 16:17:16
Parent: 1872a4ee2809a6431f8d81f50c3946a776d9c29c (AST: @_versioned types and storage are r
Branches: master, remotes/origin/master
Follows: swift-DEVELOPMENT-SNAPSHOT-2016-03-24-a
Precedes:
benchmark: Respect the SWIFT_DARWIN_XCRUN_TOOLCHAIN build setting
---------- benchmark/CMakeLists.txt ----------
index 3816dc0..2a81d7e 100644
@@ -130,7 +130,7 @@ if(NOT SWIFT_LIBRARY_PATH)
set(SWIFT_LIBRARY_PATH "${tmp_dir}/lib/swift")
endif()
-runcmd(COMMAND "xcrun" "-f" "clang"
+runcmd(COMMAND "xcrun" "-toolchain" "${SWIFT_DARWIN_XCRUN_TOOLCHAIN}" "-f" "clang"
VARIABLE CLANG_EXEC
ERROR "Unable to find Clang driver")


swift (Git)
Commit
Pull
Push
Fetch
Branch
Merge
Stash
Show in Finder
Terminal
Settings
WORKSPACE
File status
History
Search
BRANCHES
add_tool
complete-dynamicType
limited_line_ranges
master
old-swift-format
swift-format_SR-146
the_tool
TAGS
REMOTES
origin
HEAD
import-as-member
inhibit-implicit-conversions
master
master-next
revert-1513-typo-fixes-20160...
revert-2107-coregraphics
revert-2110-revert-2107-core...
rst-to-markdown
set-api
stable-swift-3.0-merge-branch
swift-2.2-branch
swift-2.2-with-migration-attri...
swift-3-indexing-model
swift-3-indexing-model-typec...
regnerjr
add_tool
cdecl-attribute
enable-ios-host-tests
master
master-next
mish-swift-3
rst-to-markdown
run-execution-tests-only
swift-2.2-branch
swift-2.2-with-migration-attri...
swift-3-api-guidelines
swift-3-indexing-model
swift-format_SR-146
All Branches
Show Remote Branches
Ancestor Order
Jump to:
Graph
Description
Commit
Author
Date
Merge pull request #2026 from practicalswift/pep8-check-for-80+-check 352f12a practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 8:31 AM
[Python] Remove unused import a75b343 practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 2:07 AM
[Python] Add "80+-check" to .pep file (needed since non-.py suffix) 99ab635 practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 2:07 AM
Merge pull request #2027 from practicalswift/gardening-20160402 b91f3d9 practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 8:02 AM
[gardening] Fix recently introduced typo: "targe" → "target" a45e7b8 practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 2:31 AM
[gardening] Fix PEP 8 regression. e8c4e1d practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 2:31 AM
Revert "Implement support for Clang's objc_runtime_visible attribute." 09973e6 Ted Kremenek <kremenek@apple.com> Apr 2, 2016, 7:33 AM
Merge pull request #2016 from practicalswift/gardening-20160401 14197f6 Dmitri Gribenko <gribozavr@gmail.com> Apr 2, 2016, 1:58 AM
[gardening] Fix PEP 8 regression. 292bbe9 practicalswift <practicalswift@users.noreply.gith... Apr 1, 2016, 2:14 PM
[gardening] Fix recently introduced typo: "referenece" → "reference" c196a44 practicalswift <practicalswift@users.noreply.gith... Apr 1, 2016, 2:14 PM
[gardening] Fix file header formatting. a7acb1a practicalswift <practicalswift@users.noreply.gith... Apr 1, 2016, 2:14 PM
[gardening] Fix recently introduced typo: "althouth" → "although" fa1d5d2 practicalswift <practicalswift@users.noreply.gith... Apr 1, 2016, 2:14 PM
Merge pull request #2018 from practicalswift/remove-crashes-without-apache-license-header 99b89d3 practicalswift <practicalswift@users.noreply.gith... Apr 2, 2016, 12:14 AM
[crashers] Remove crash cases without explicit Apache license header. 6a3af35 practicalswift <practicalswift@users.noreply.gith... Apr 1, 2016, 2:48 PM
Implement support for Clang's objc_runtime_visible attribute. 2c1f19a Doug Gregor <dgregor@apple.com> Apr 1, 2016, 11:07 PM
Merge pull request #2024 from gottesmm/add-80+-check f9ef148 Michael Gottesman <gottesmm@users.noreply.gi... Apr 1, 2016, 10:30 PM
Add a new utility called 80+-check. b1ffbf8 Michael Gottesman <mgottesman@apple.com> Apr 1, 2016, 10:26 PM
Sorted by path
Search
include
swift
IDE
CodeCompletion.h
include/swift/IDE/CodeCompletion.h
Hunk 1 : Lines 7-20
Reverse hunk
//
// See http://swift.org/LICENSE.txt for license information
// See http://swift.org/CONTRIBUTORS.txt for the list of Swift project authors
//
//===--===//
-#ifndef SWIFT_IDE_CODE_COMPLETION_H
-#define SWIFT_IDE_CODE_COMPLETION_H
+#ifndef SWIFT_IDE_CODECOMPLETION_H
+#define SWIFT_IDE_CODECOMPLETION_H
#include "swift/AST/Identifier.h"
#include "swift/Basic/LLVM.h"
#include "llvm/ADT/ArrayRef.h"
#include "llvm/ADT/StringRef.h"
#include "llvm/Support/Allocator.h"
Hunk 2 : Lines 43-59
Reverse hunk
///
/// \code
/// code-completion-token:
///   '#^' identifier '^#'
/// \endcode
///
-/// \param Input test source code
-/// \param TokenName names the token which position should be returned in
+/// \param Input test source code.
+/// \param TokenName names the token whose position should be returned in
/// \p CompletionOffset.
/// \param CompletionOffset set to ~0U on error, or to a 0-based byte offset on
-/// success
+/// success.
///
/// \returns test source code without any code completion tokens.
std::string removeCodeCompletionTokens(StringRef Input,
                                       StringRef TokenName,
                                       unsigned *CompletionOffset);
Merge pull request #1987 from danra/danra-patch-1
Cosmetic fixes (NFC)
Commit: 2ab38e0660ae8fc6751725d0303edca9f46fdaa9 ...
Parents: 65ea5c7e56, 71f354e151
Author: Chris Lattner <lattner@users.noreply.github.com>
Date: March 31, 2016 at 8:45:05 PM PDT
Labels: regnerjr/master

“Commits seem cool. How do I make one?”

–You

`git commit`

but it is actually more complicated than that

Remember Git is stupid

- Can’t commit because it does not know what you want to save.
- You need to tell it manually which things are important and which are not.
- Helps you not save stuff that was changed in error.

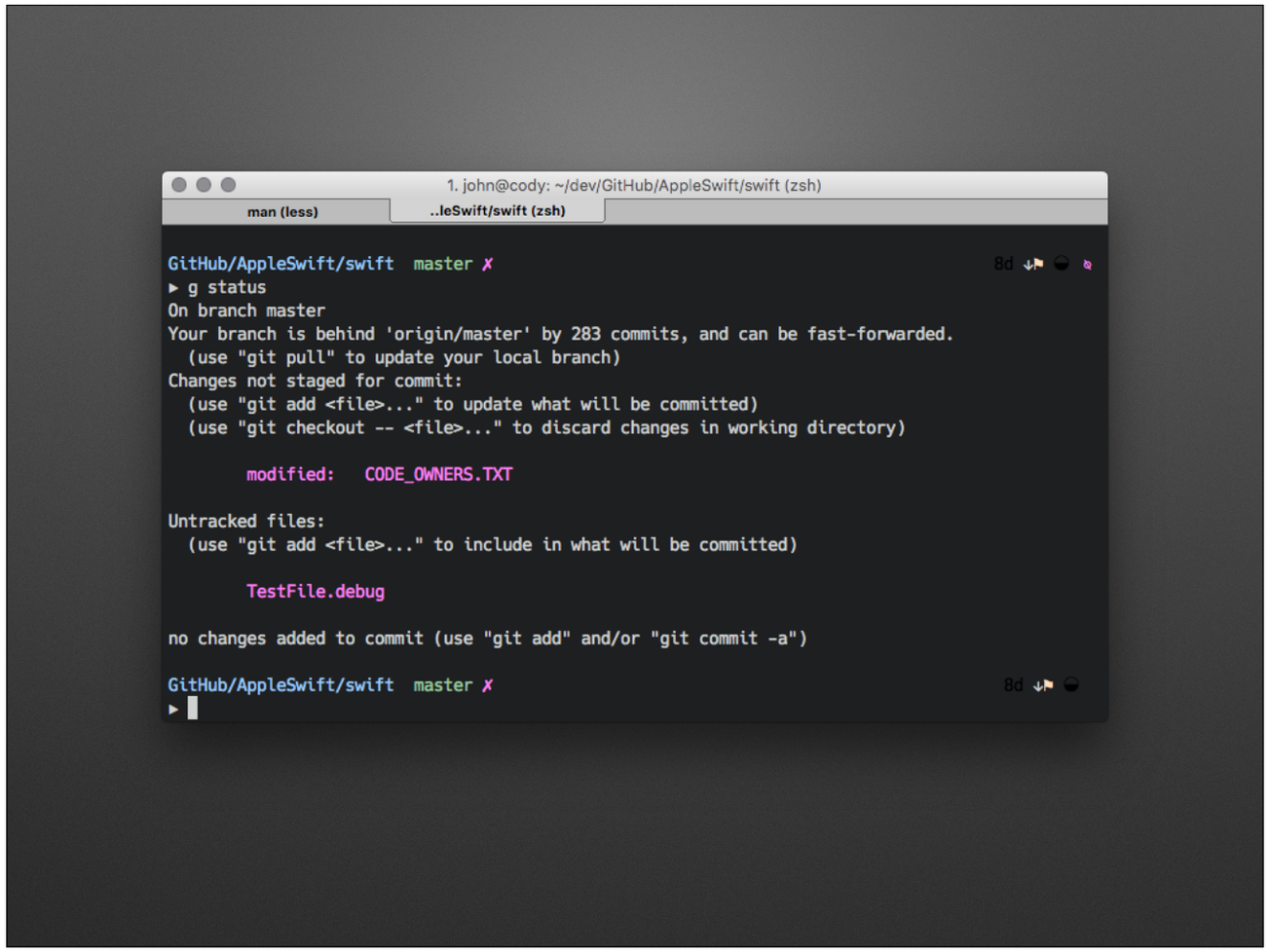
`git commit` but it is actually a bit more complex than that. And this is probably where most people want to quit.

You have to decide, by hand manually, By dragging files or clicking buttons or typing commands

What do you really want to be part of your changes? I often put red boxes everywhere when debugging layouts, I don't want those saved in my projects history.

`git status`

Remind me. What I am doing here again?

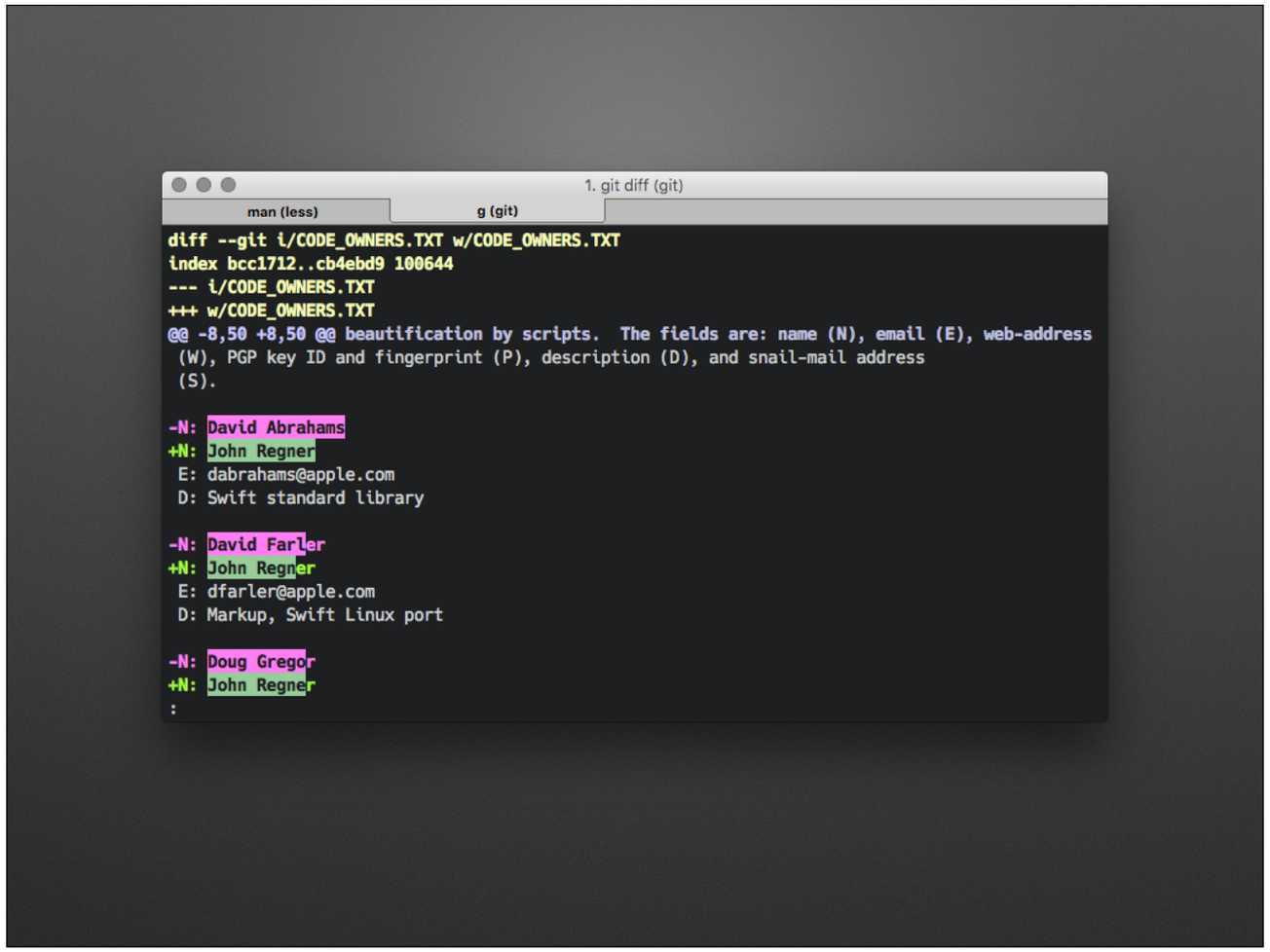


Did you add a new file, or change existing ones or both?
Do you remember what you did? I almost never do.

`git status` is your friend. These visual tools will usually show your status by default.
There you can decide what you want to do.

`git diff`

I changed that file? What did I do again?


1. git diff (git)
man (less)
g (git)
diff --git i/CODE_OWNERS.TXT w/CODE_OWNERS.TXT
index bcc1712..cb4ebd9 100644
--- i/CODE_OWNERS.TXT
+++ w/CODE_OWNERS.TXT
@@ -8,50 +8,50 @@ beautification by scripts. The fields are: name (N), email (E), web-address
(W), PGP key ID and fingerprint (P), description (D), and snail-mail address
(S).
-N: David Abrahams
+N: John Regner
E: dabrahams@apple.com
D: Swift standard library
-N: David Farler
+N: John Regner
E: dfarler@apple.com
D: Markup, Swift Linux port
-N: Doug Gregor
+N: John Regner
:

All of this has been a sideshow

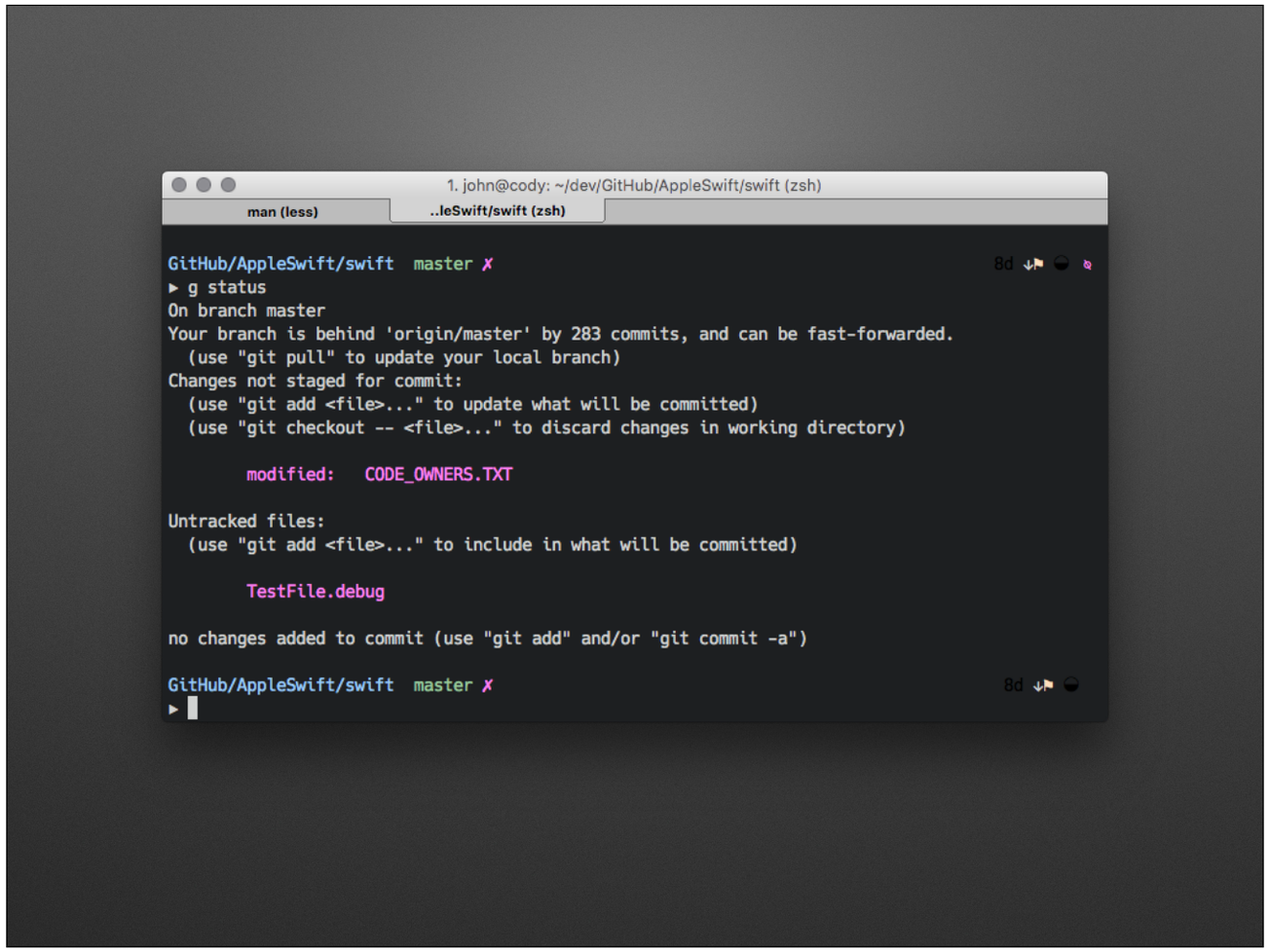
- We still want to make a new commit.
- We can see which files changed with `git status`
- We can see the actual changes with `git diff`
- Now we want to save some of them.

`git add`

Add this to the “staging area”
I am ready to commit these changes.

`git checkout`

Undo the changes to this file.
Checkout the saved version of this file.

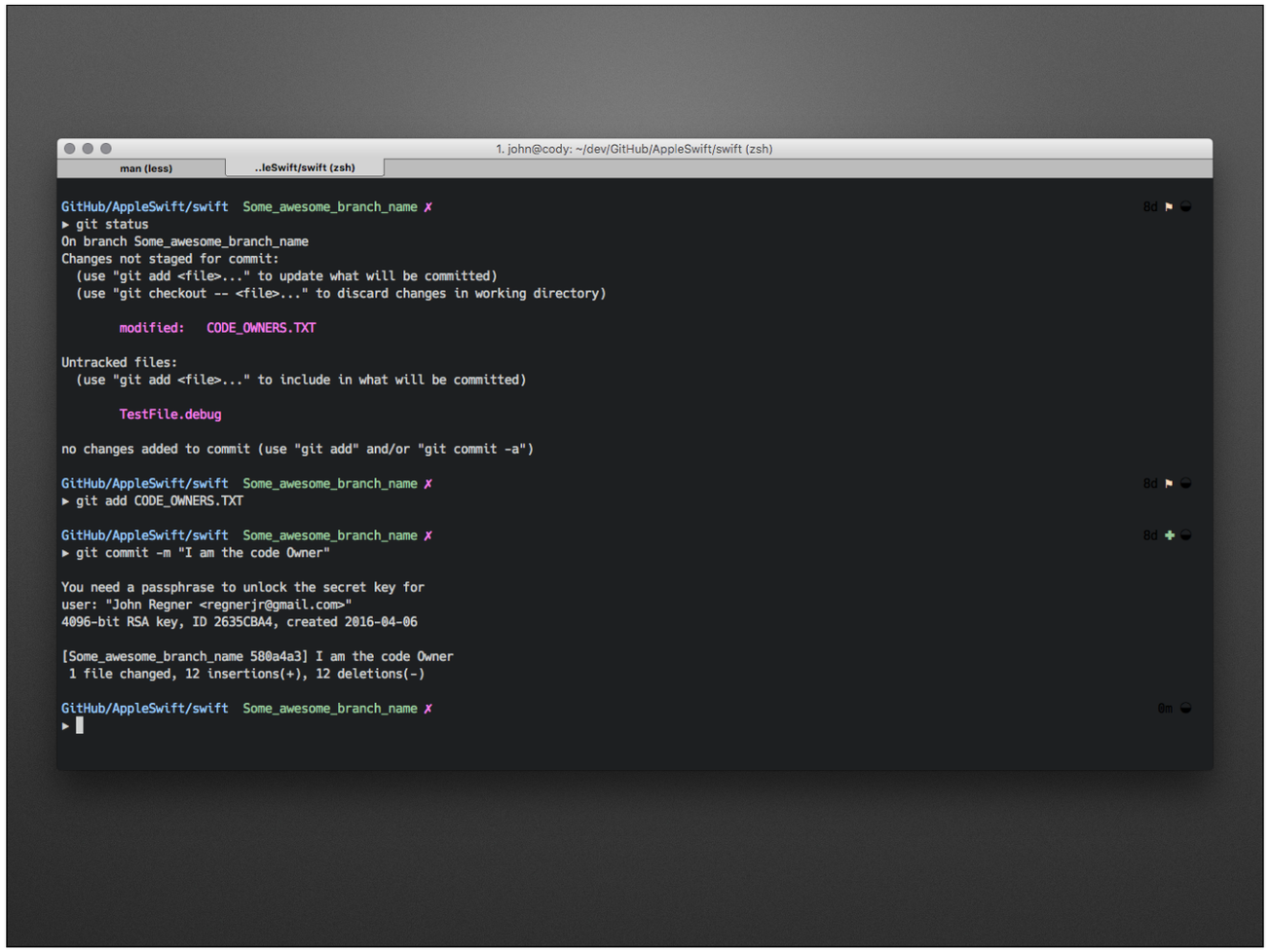


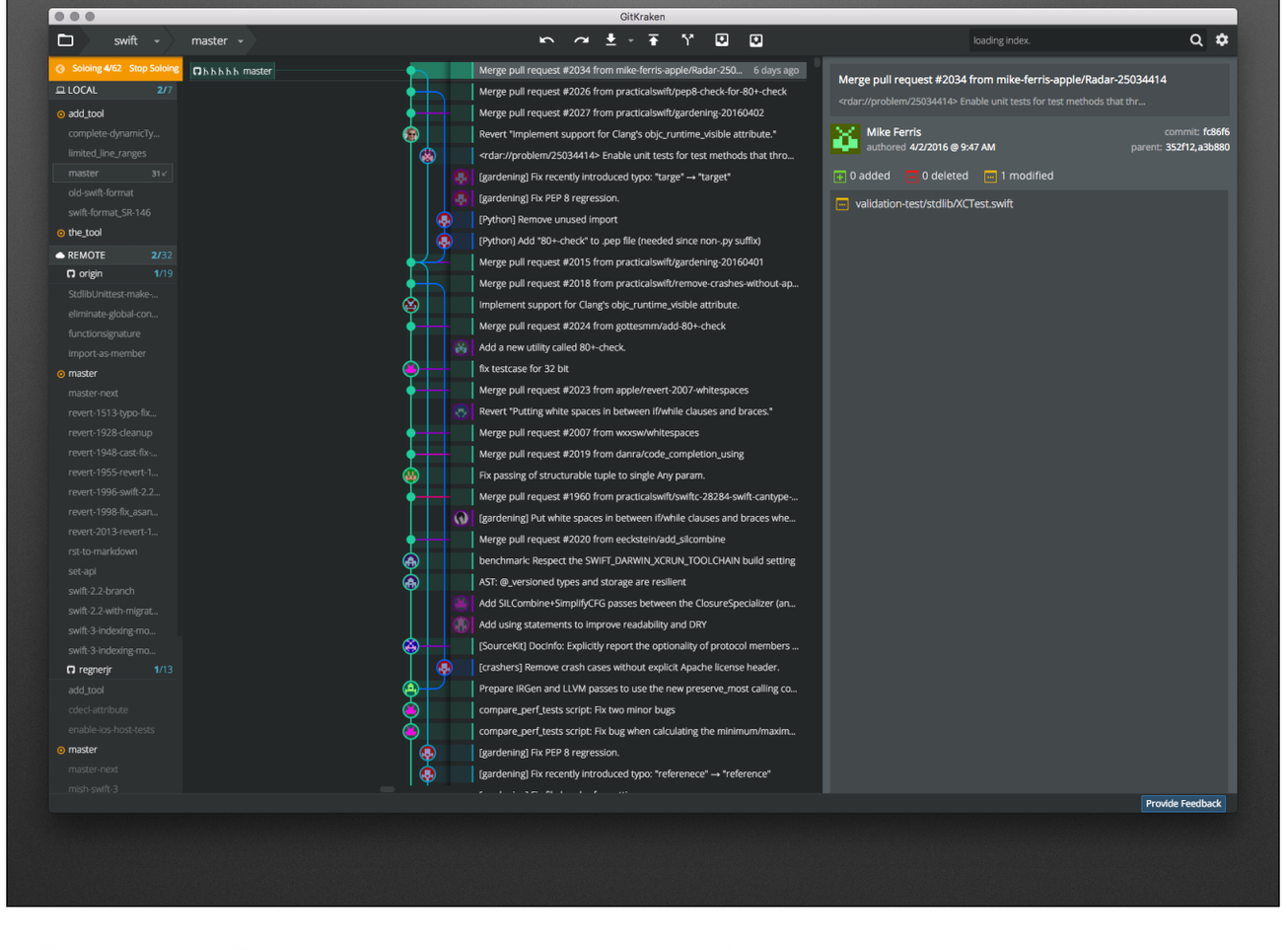
Probably look at the status again to make sure everything is as you like it.

Then you are ready to commit.

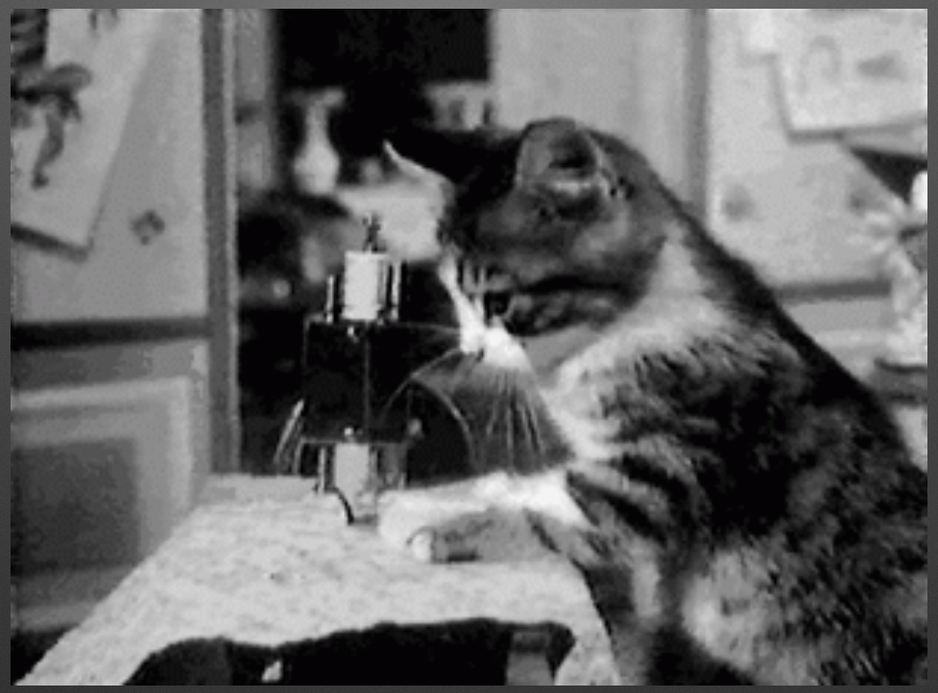
`git commit`

Now that we have “added” the changes we want to save.
We can commit our changes.


1. john@cody: ~/dev/GitHub/AppleSwift/swift (zsh)
man (less)
..leSwift/swift (zsh)
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 8d
▶ git status
On branch Some_awesome_branch_name
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
modified: CODE_OWNERS.TXT
Untracked files:
(use "git add <file>..." to include in what will be committed)
TestFile.debug
no changes added to commit (use "git add" and/or "git commit -a")
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 8d
▶ git add CODE_OWNERS.TXT
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 8d
▶ git commit -m "I am the code Owner"
You need a passphrase to unlock the secret key for
user: "John Regner <regnerjr@gmail.com>"
4096-bit RSA key, ID 2635CBA4, created 2016-04-06
[Some_awesome_branch_name 580a4a3] I am the code Owner
1 file changed, 12 insertions(+), 12 deletions(-)
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 0m
▶



Hooray You've moved the project forward. This is huge. Be proud. We had to know a ton of commands to get this far!!!!



This is awesome. I know how to use git.

What if I mess up?

Use git's eject button.

`git reset —hard`

Takes you back to the previous commit. And cleans everything.

Sometimes it is nice to think of `git` as a bit of a safety net. You can try things out knowing that you can always get back.

`git reset --hard` is you eject button. Something went horribly wrong, I just want to go back to my last snapshot.

Intermediate Git

Do we need a stretch break?

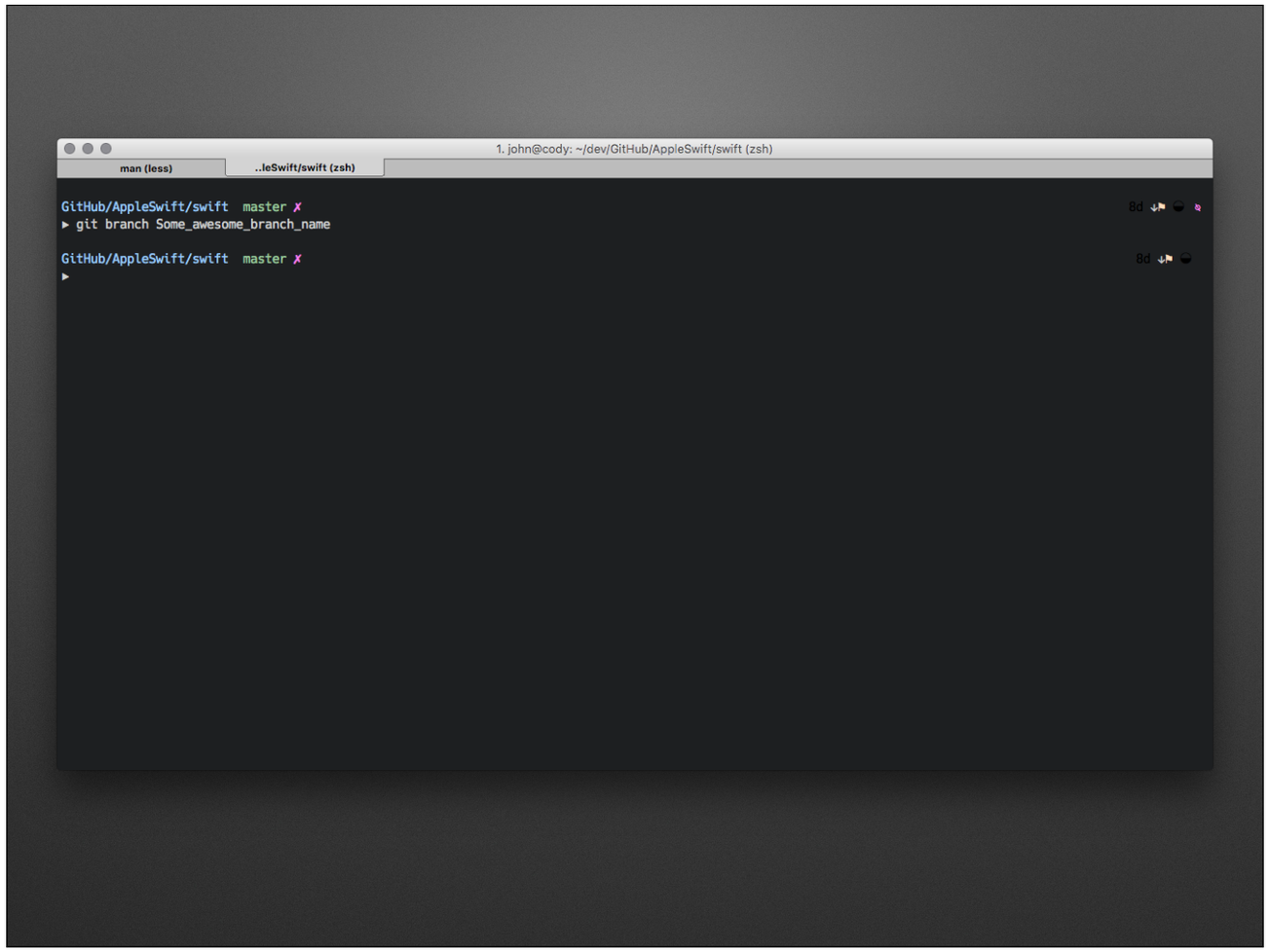
Get excited I made cool pictures!!!!!!!!!

Branching

Branches are awesome
And fast
Make lots of them
Make them often

`git branch <name>`

I can name it whatever I want?



Everything is a branch. Most of your initial project work will be commits added to a branch called master.
Branches basically point to one commit and all of their parent commits. Just like branches of a tree.

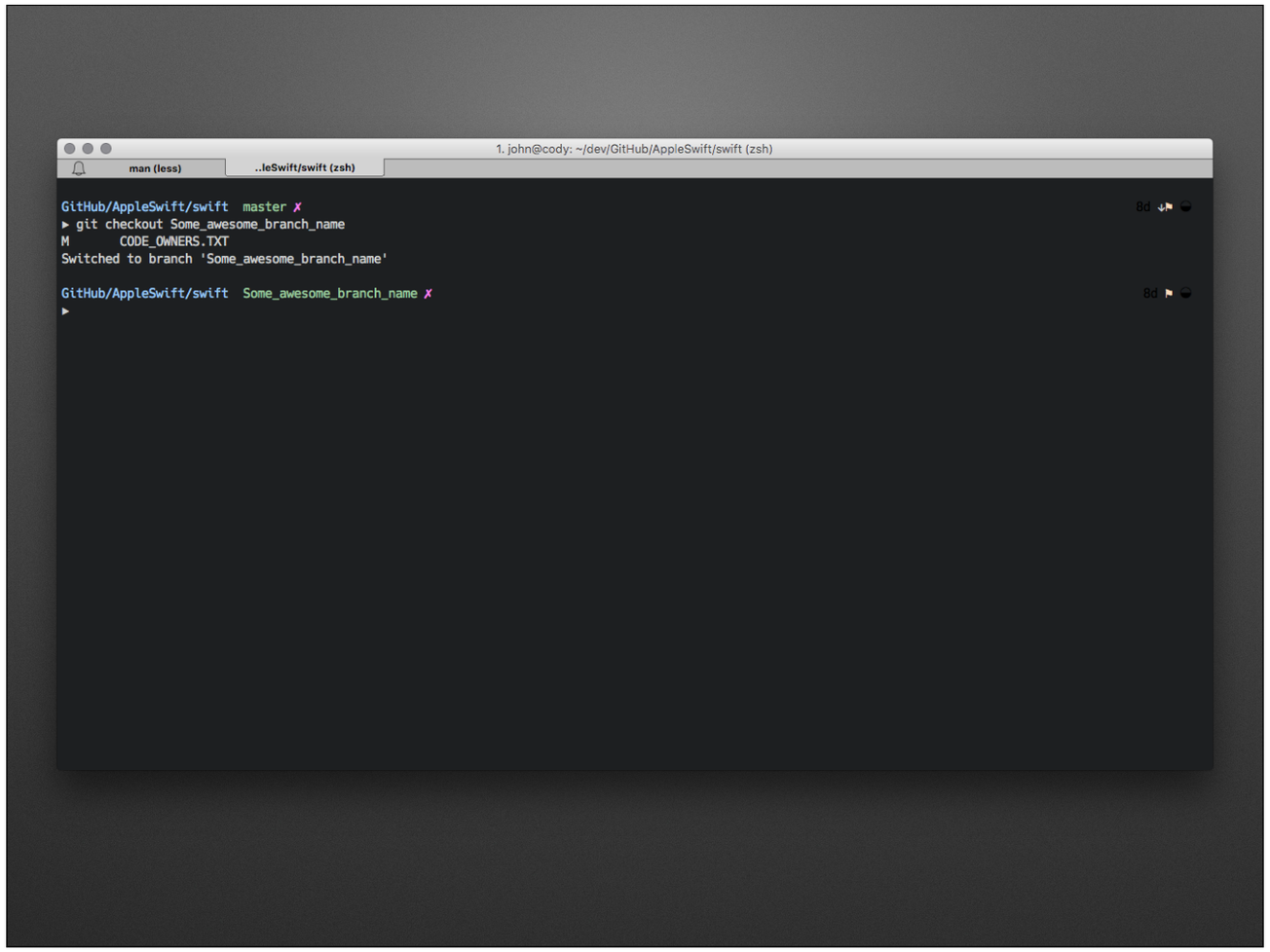
`git branch Some_awesome_branch_name` to make a new one.

`git checkout <name>`

This is how you change branches.

`git checkout Some_awesome_branch_name` to switch to that branch.

Continue fixing bugs, adding features. Depending on your team, You will probably do one fix or one feature per branch.



Notice how my changed files came with me!

Merging

Bring branches together

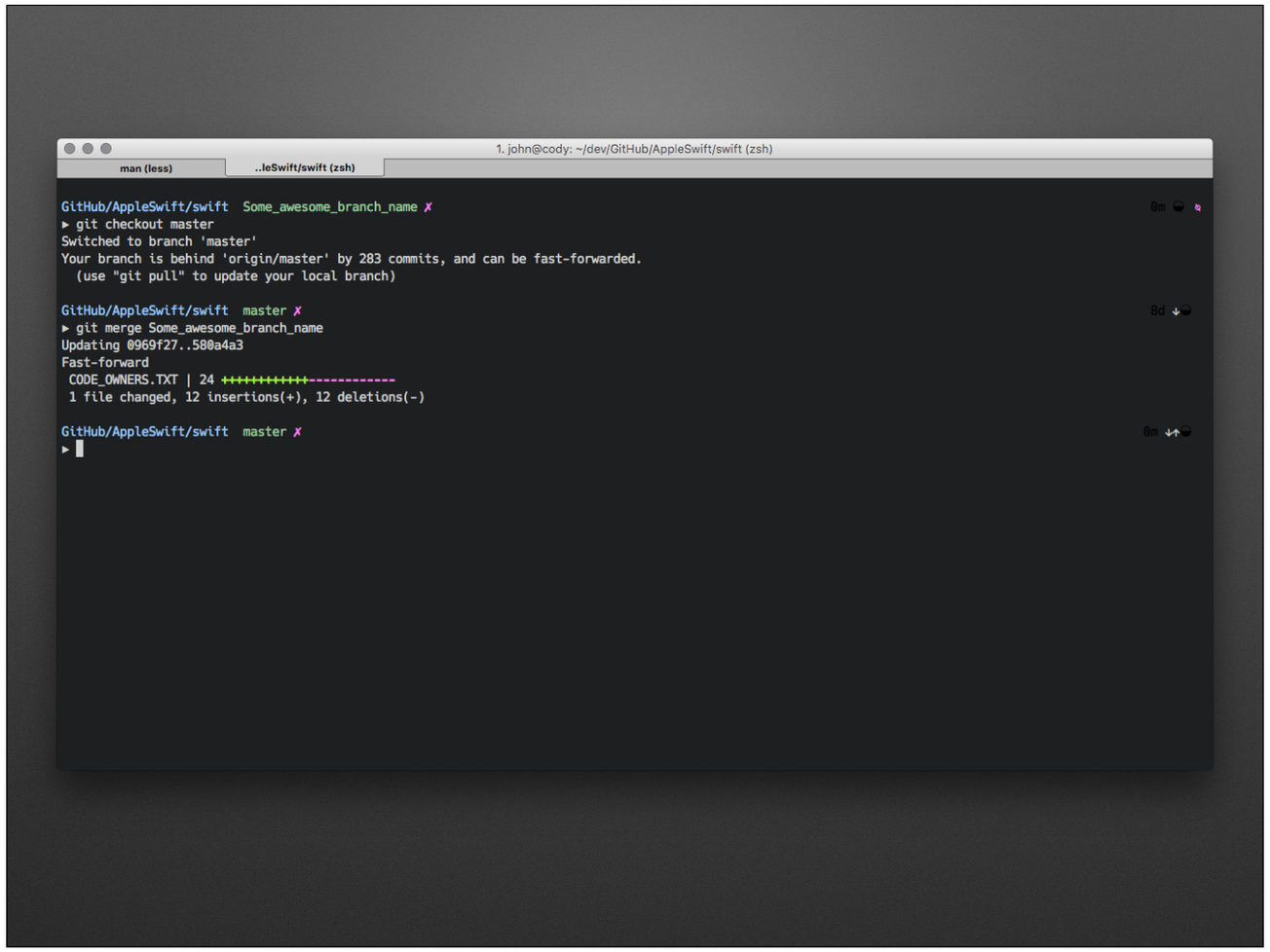
This will happen on the Github website a lot.

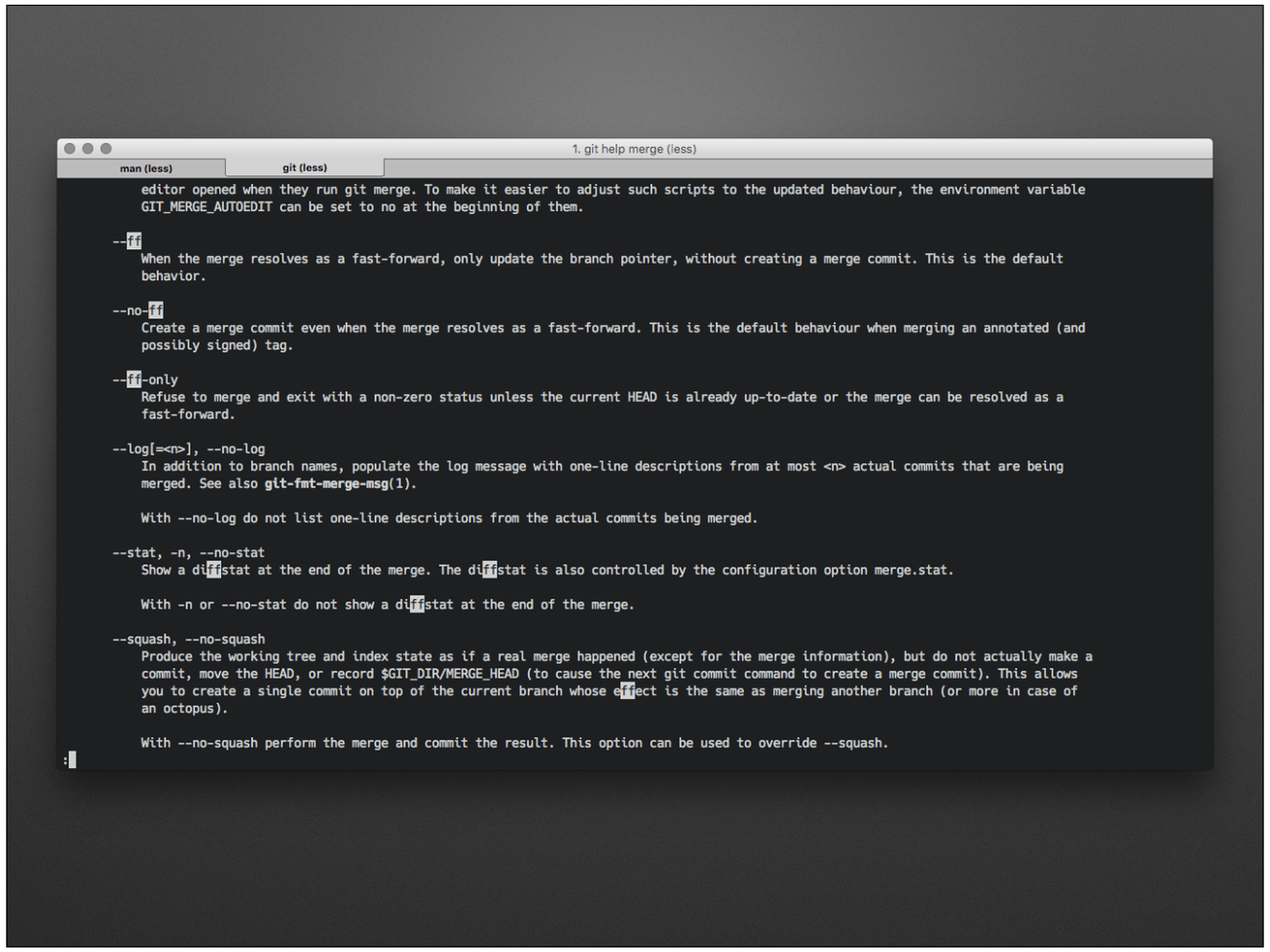
`git merge <name>`

Merge the changes from branch name into your current branch.

Move to the destination branch. Then just `git merge branch_with_changes_you_want`

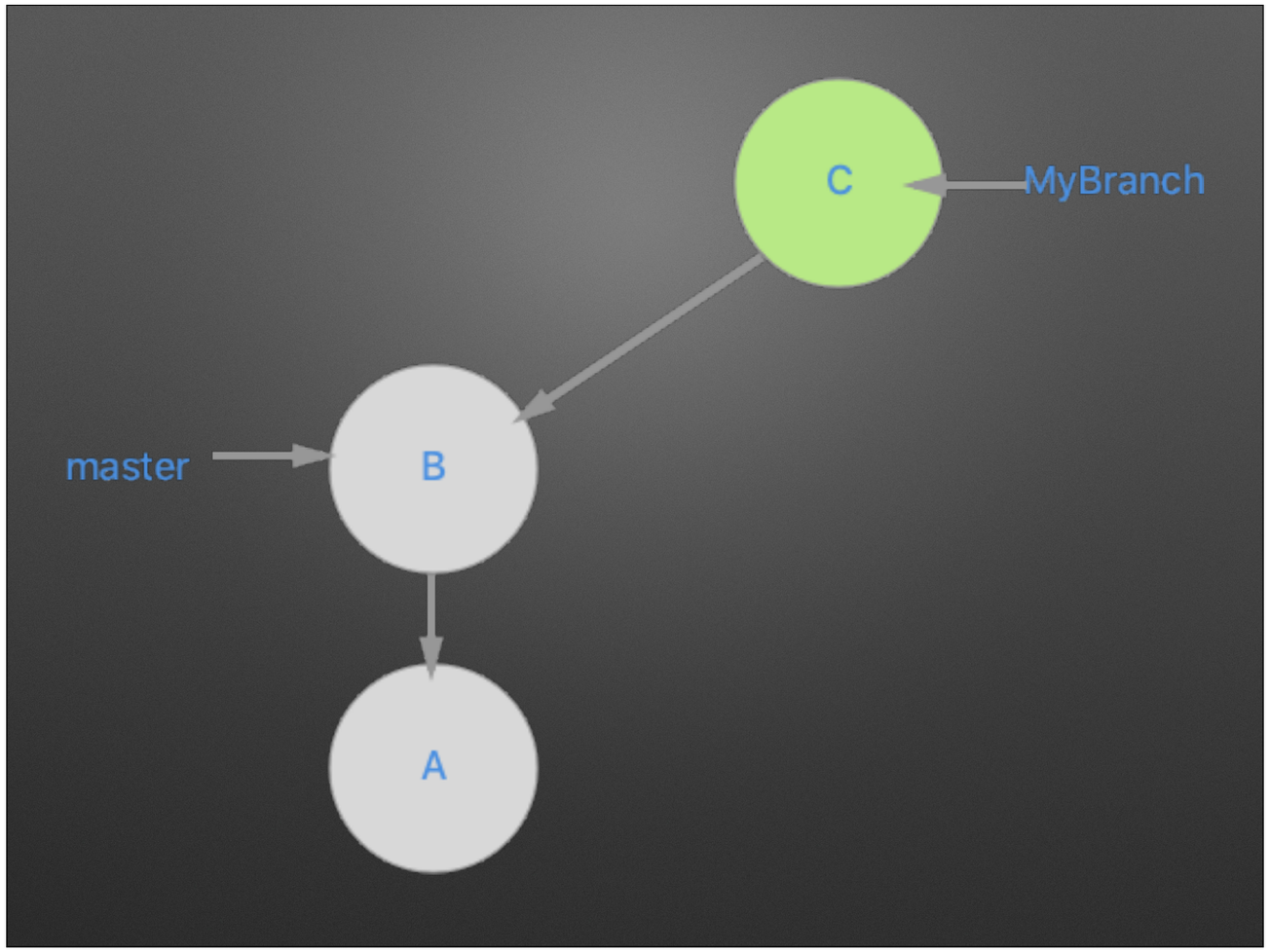
You will not use this that often if using a GitHub workflow with pull requests, because there is a merge button on the web interface.
But if there are issues the web interface will tell you to do it this way.


1. john@cody: ~/dev/GitHub/AppleSwift/swift (zsh)
man (less)
..leSwift/swift (zsh)
GitHub/AppleSwift/swift Some_awesome_branch_name ✗
0m
▶ git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 283 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
GitHub/AppleSwift/swift master ✗
8d
▶ git merge Some_awesome_branch_name
Updating 0969f27..580a4a3
Fast-forward
CODE_OWNERS.TXT | 24 ++++++++++-----------
1 file changed, 12 insertions(+), 12 deletions(-)
GitHub/AppleSwift/swift master ✗
0m
▶



All about source management and specifically what do you want your history to look like.
Probably more of a senior level decision make sure you have your local system set up to do what your team expects.

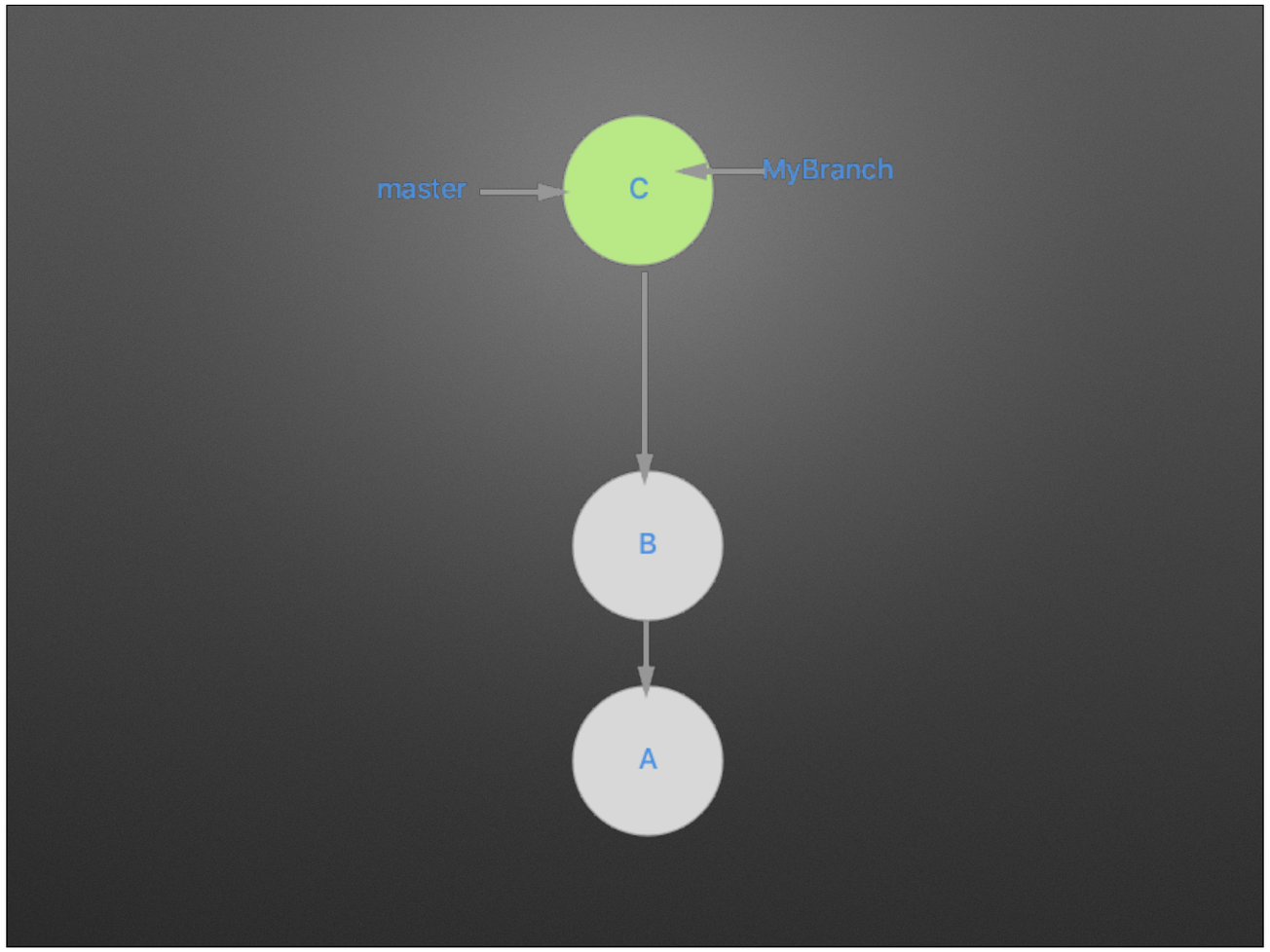
Talk through the pictures.



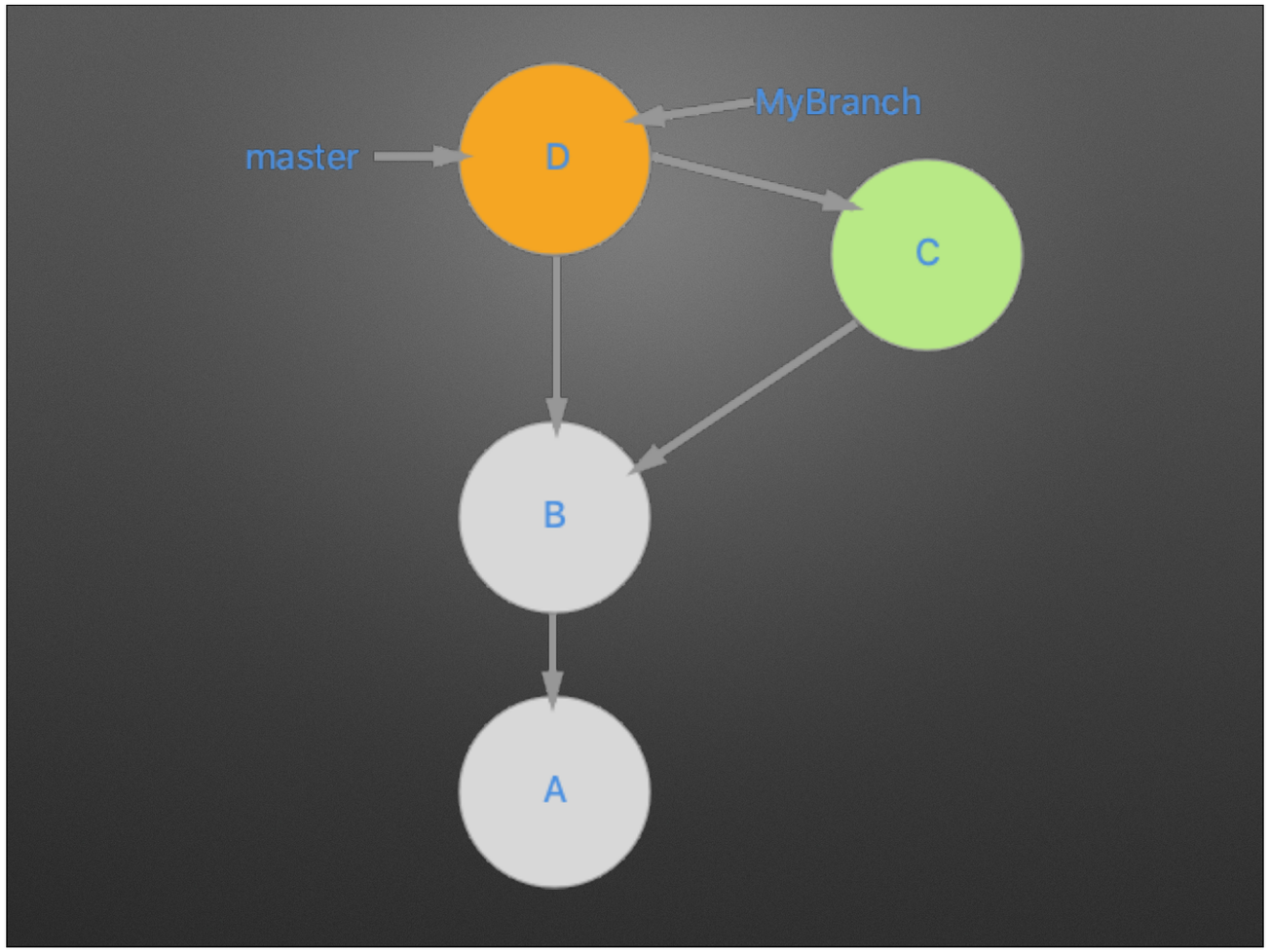
The little circle are commits. and the names are branches.

This is actually how git tracks stuff, branches are just labels that point to commits.

Commits point to their parent.

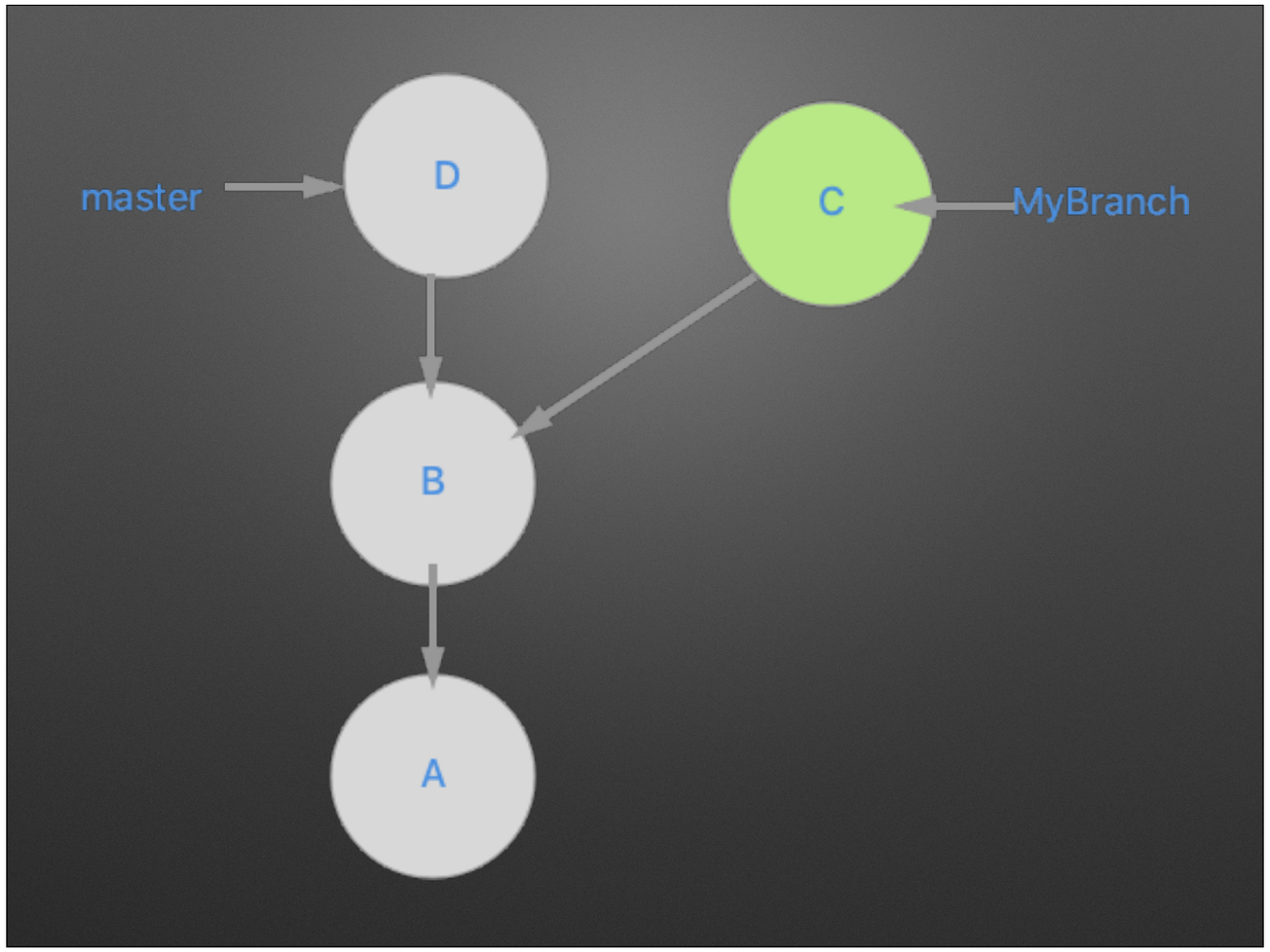


This is a fast forward merge. This is the default behavior. We really just moved the master pointer to point to C instead of B



Some developers really like this new merge commit.
It does seem a bit more true to what happened.
But there is a beauty in having a totally linear history.

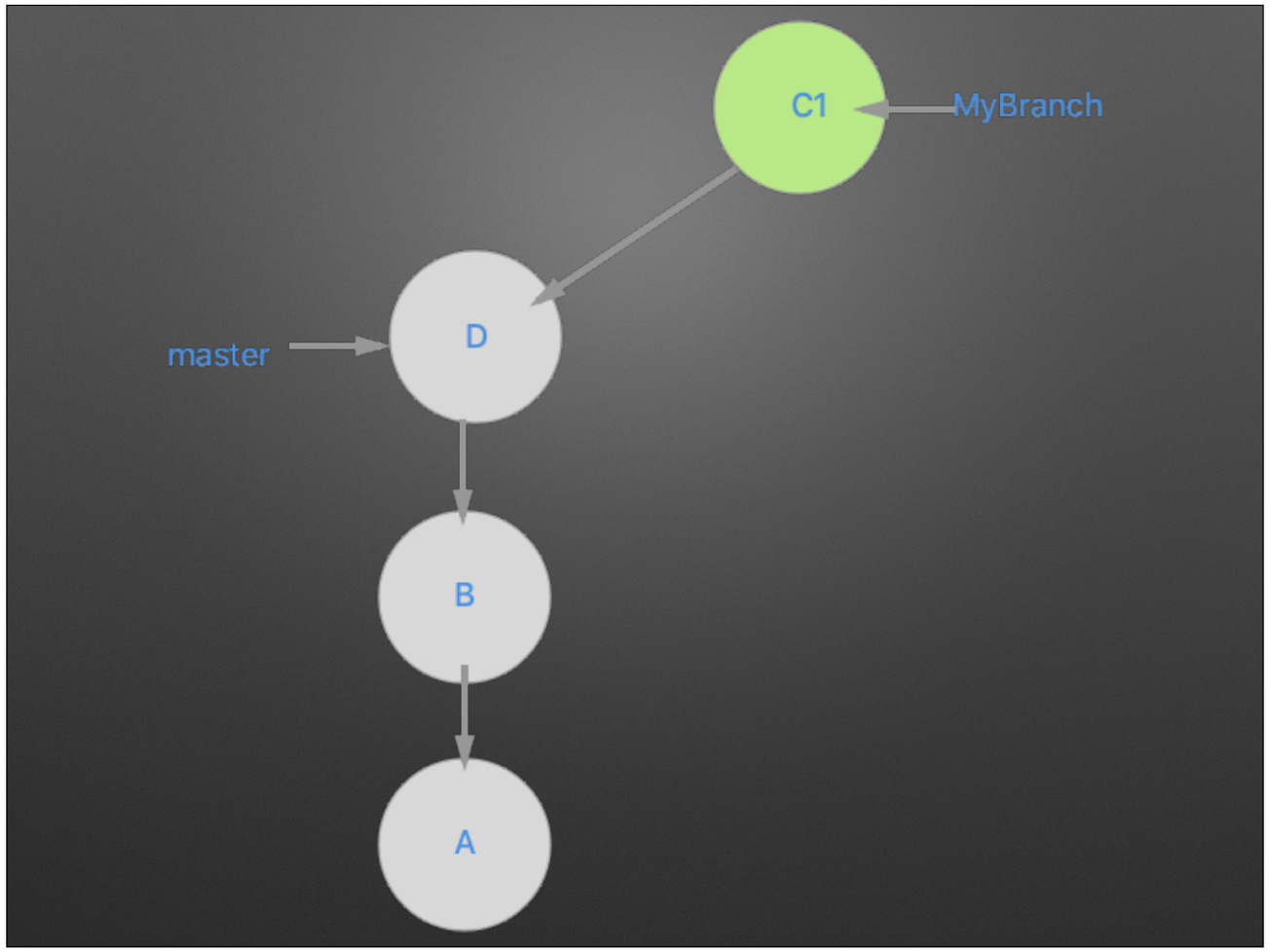
How do I FF when other developers have made changes



If I try to do a fast forward merge, Git will not let me since the histories have diverged. I need to bring them together again. so that the master is pointing at a parent of C. I need to change the base of MyBranch.

I need to rebase

`git rebase master`



I have successfully rebased on top of master.

Now I can FF merge and have that beautiful linear history.

This will be requested of you often when working on Open Source Projects.
Especially ones with a lot of changes.

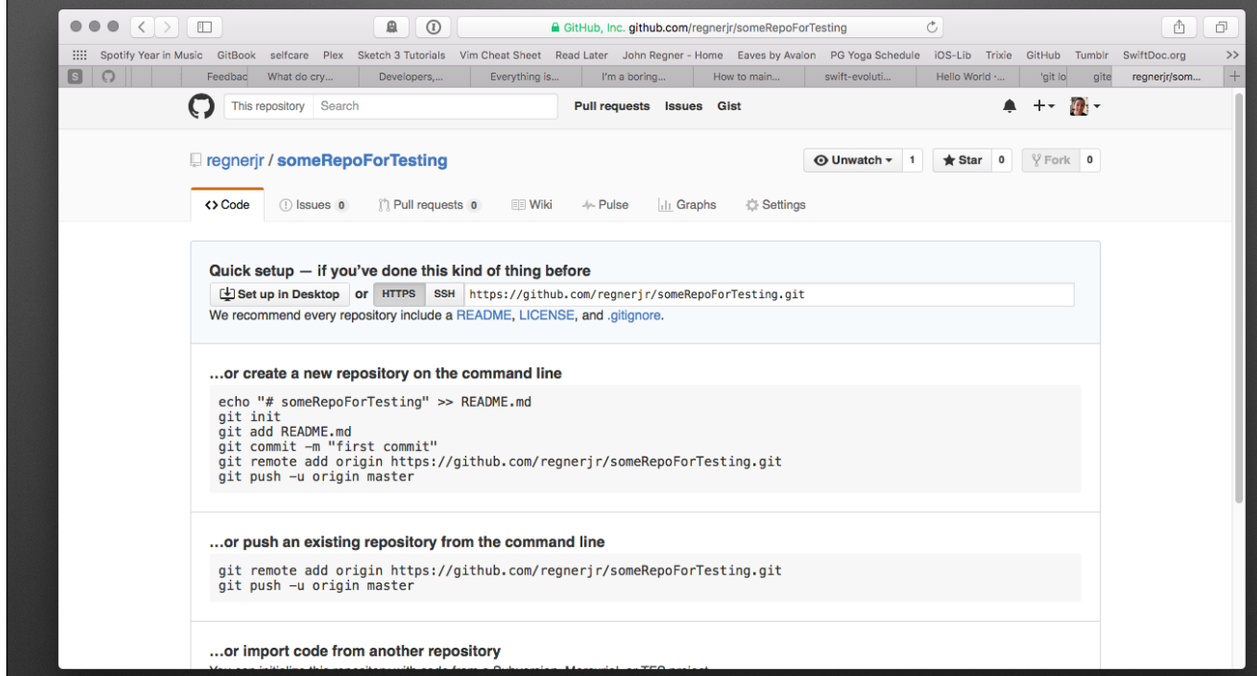
Look How much
we can do
without GitHub

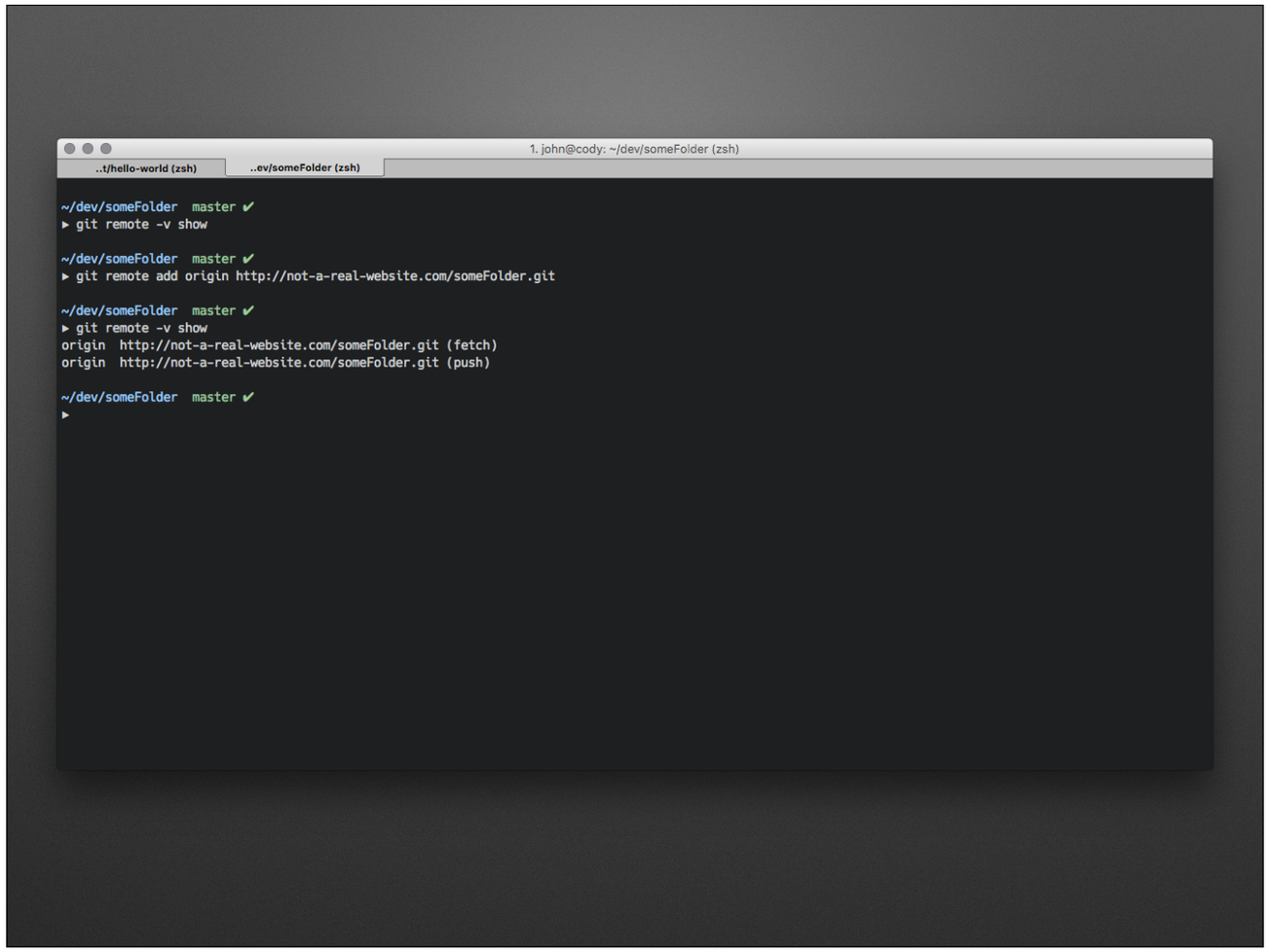
GitHub - Social Coding

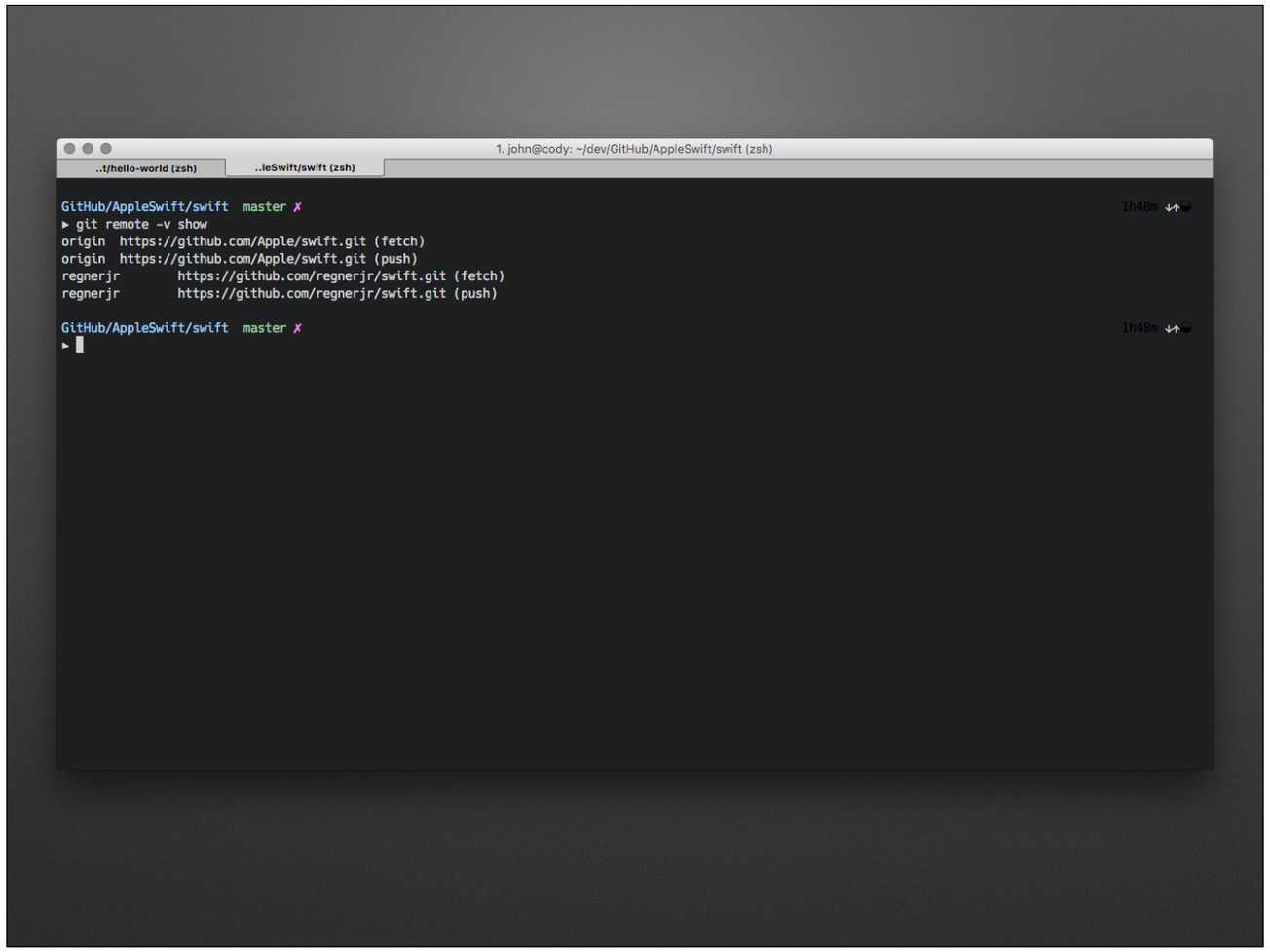
A website for interacting with friends and code

git remote add

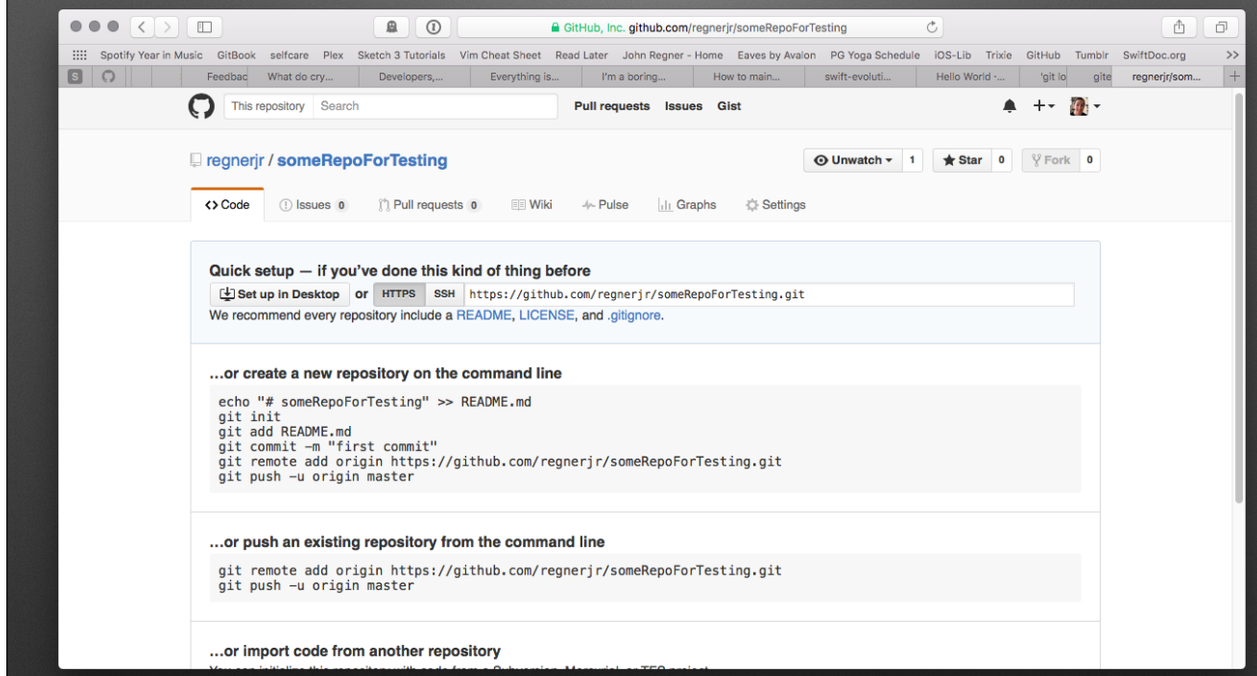
Connect your local code to GitHub (or any other git server)


GitHub, Inc. github.com/regnerjr/someRepoForTesting
This repository
Search
Pull requests
Issues
Gist
regnerjr / someRepoForTesting
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/regnerjr/someRepoForTesting.git
We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# someRepoForTesting" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/regnerjr/someRepoForTesting.git
git push -u origin master
...or push an existing repository from the command line
git remote add origin https://github.com/regnerjr/someRepoForTesting.git
git push -u origin master
...or import code from another repository


1. john@cody: ~/dev/someFolder (zsh)
..t/hello-world (zsh)
..ev/someFolder (zsh)
~/dev/someFolder master ✔
▶ git remote -v show
~/dev/someFolder master ✔
▶ git remote add origin http://not-a-real-website.com/someFolder.git
~/dev/someFolder master ✔
▶ git remote -v show
origin http://not-a-real-website.com/someFolder.git (fetch)
origin http://not-a-real-website.com/someFolder.git (push)
~/dev/someFolder master ✔
▶


1. john@cody: ~/dev/GitHub/AppleSwift/swift (zsh)
..t/hello-world (zsh)
..leSwift/swift (zsh)
GitHub/AppleSwift/swift master ✗
1h48m
▶ git remote -v show
origin https://github.com/Apple/swift.git (fetch)
origin https://github.com/Apple/swift.git (push)
regnerjr https://github.com/regnerjr/swift.git (fetch)
regnerjr https://github.com/regnerjr/swift.git (push)
GitHub/AppleSwift/swift master ✗
1h48m
▶

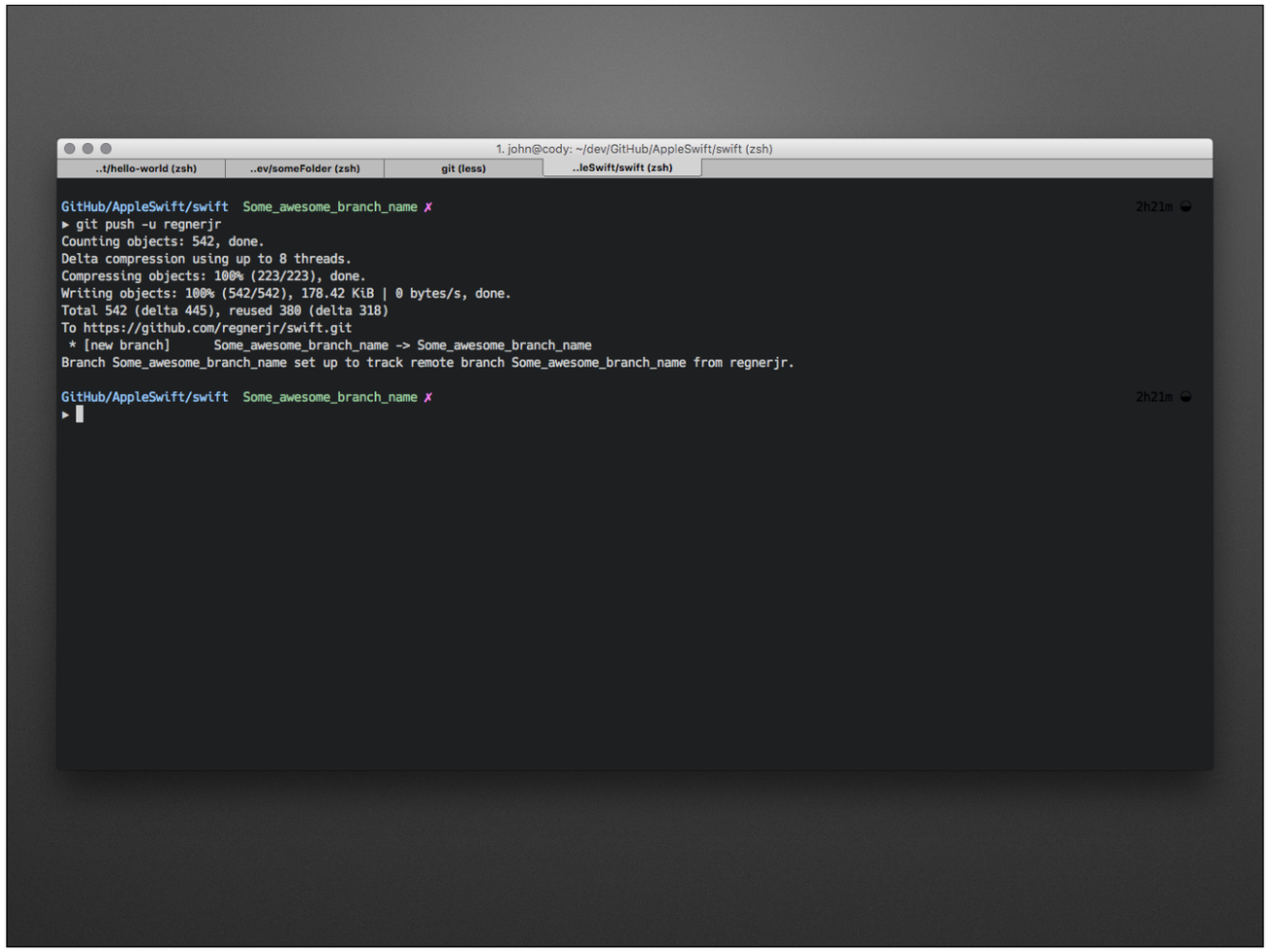

THAT
SOUNDS
HARD!


GitHub, Inc. github.com/regnerjr/someRepoForTesting
This repository
Search
Pull requests
Issues
Gist
regnerjr / someRepoForTesting
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/regnerjr/someRepoForTesting.git
We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# someRepoForTesting" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/regnerjr/someRepoForTesting.git
git push -u origin master
...or push an existing repository from the command line
git remote add origin https://github.com/regnerjr/someRepoForTesting.git
git push -u origin master
...or import code from another repository

`git push`

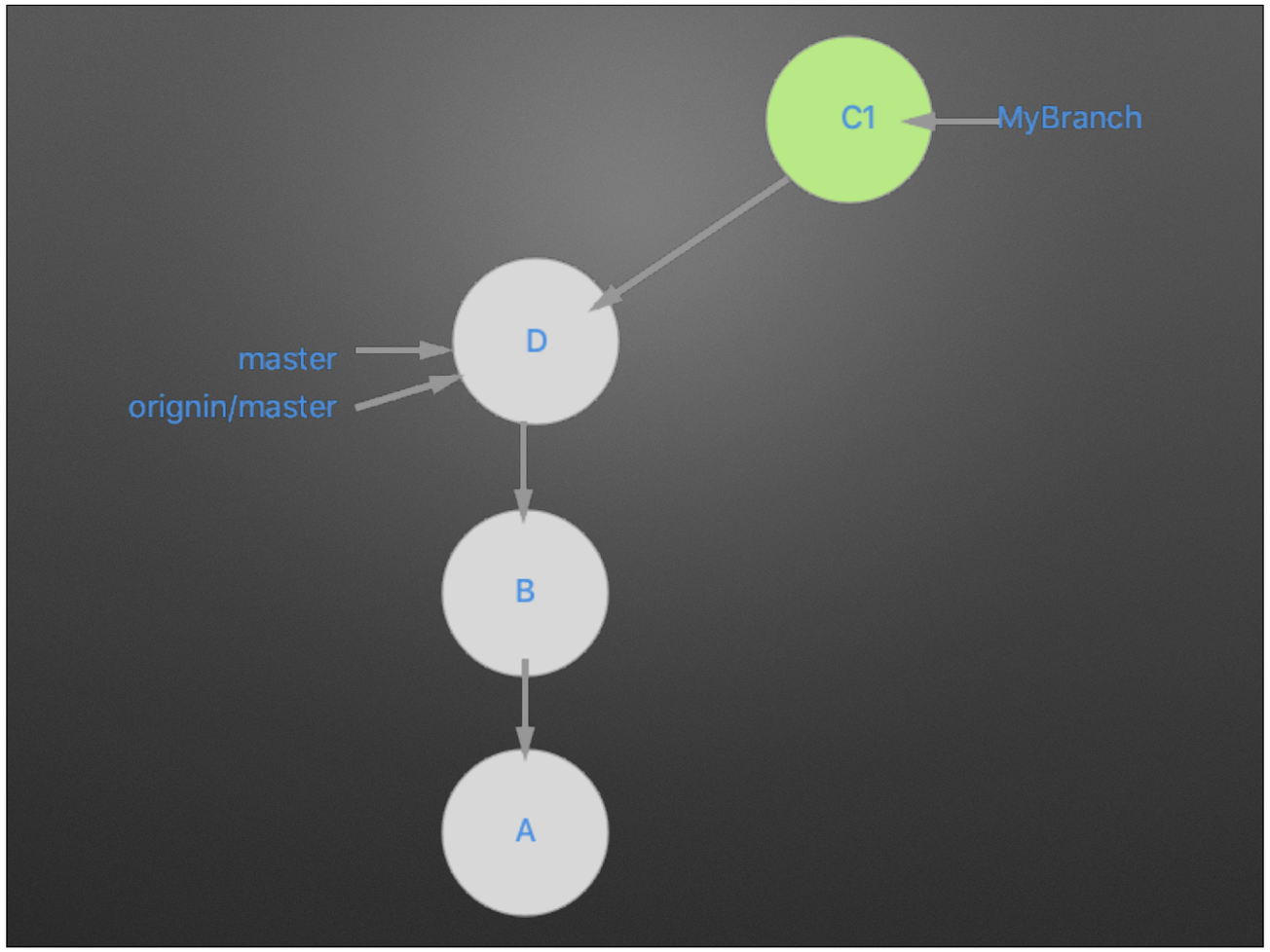
There are details here that are confusing.
Usually you can just `git push`

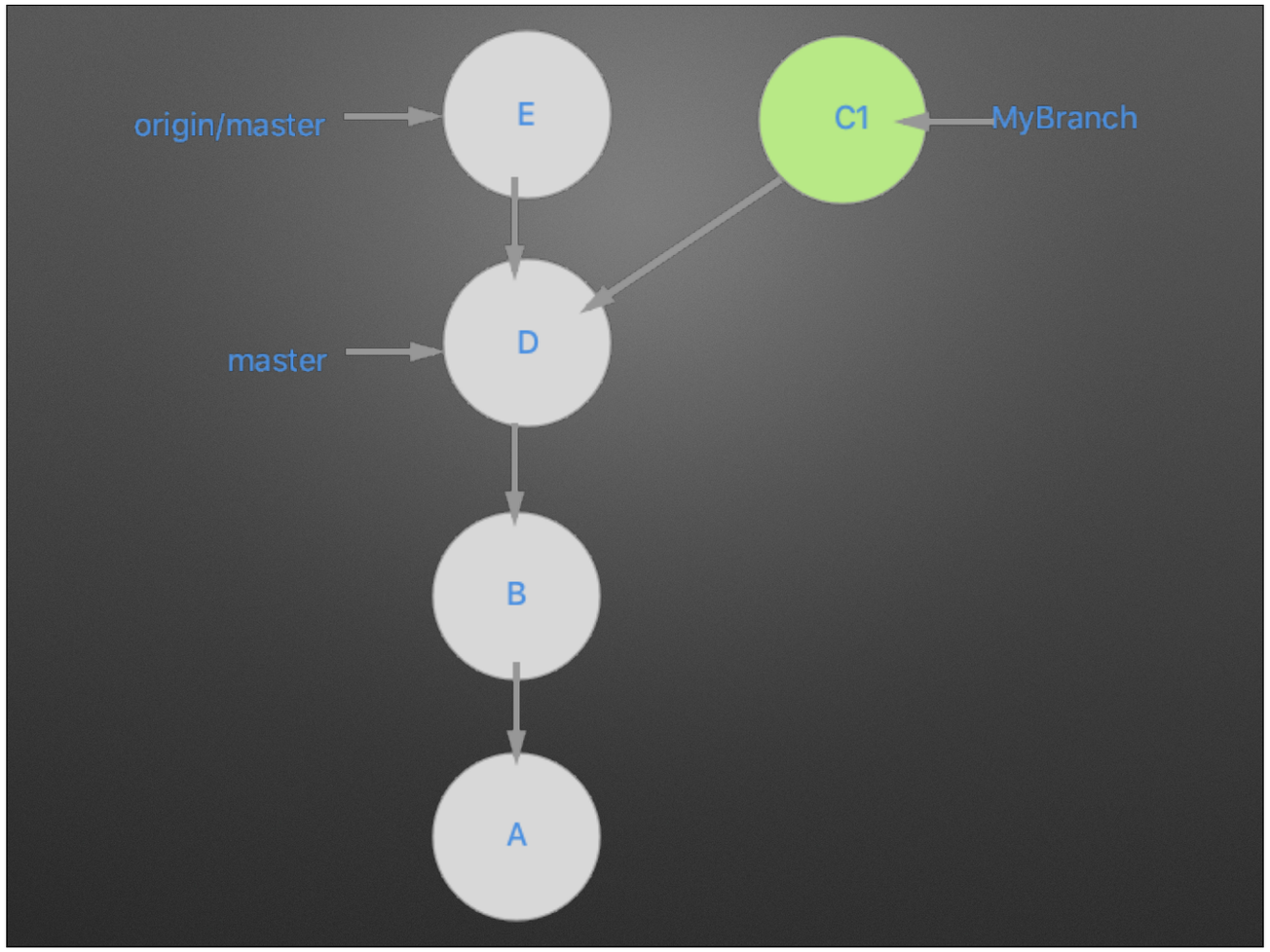
Puts you code up on the github website to share with friends, and co-workers and everybody!!! Social!!!


1. john@cody: ~/dev/GitHub/AppleSwift/swift (zsh)
..t/hello-world (zsh)
..ev/someFolder (zsh)
git (less)
..leSwift/swift (zsh)
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 2h21m
▶ git push -u regnerjr
Counting objects: 542, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (223/223), done.
Writing objects: 100% (542/542), 178.42 KiB | 0 bytes/s, done.
Total 542 (delta 445), reused 380 (delta 318)
To https://github.com/regnerjr/swift.git
* [new branch] Some_awesome_branch_name -> Some_awesome_branch_name
Branch Some_awesome_branch_name set up to track remote branch Some_awesome_branch_name from regnerjr.
GitHub/AppleSwift/swift Some_awesome_branch_name ✗ 2h21m
▶

`git fetch —all`

Get all remote changes.
Bring them down to my repo so I can see them


C1
MyBranch
D
master
orignin/master
B
A


origin/master
E
C1
MyBranch
master
D
B
A

`git pull`

Really just `git fetch` and `git merge` together in one.

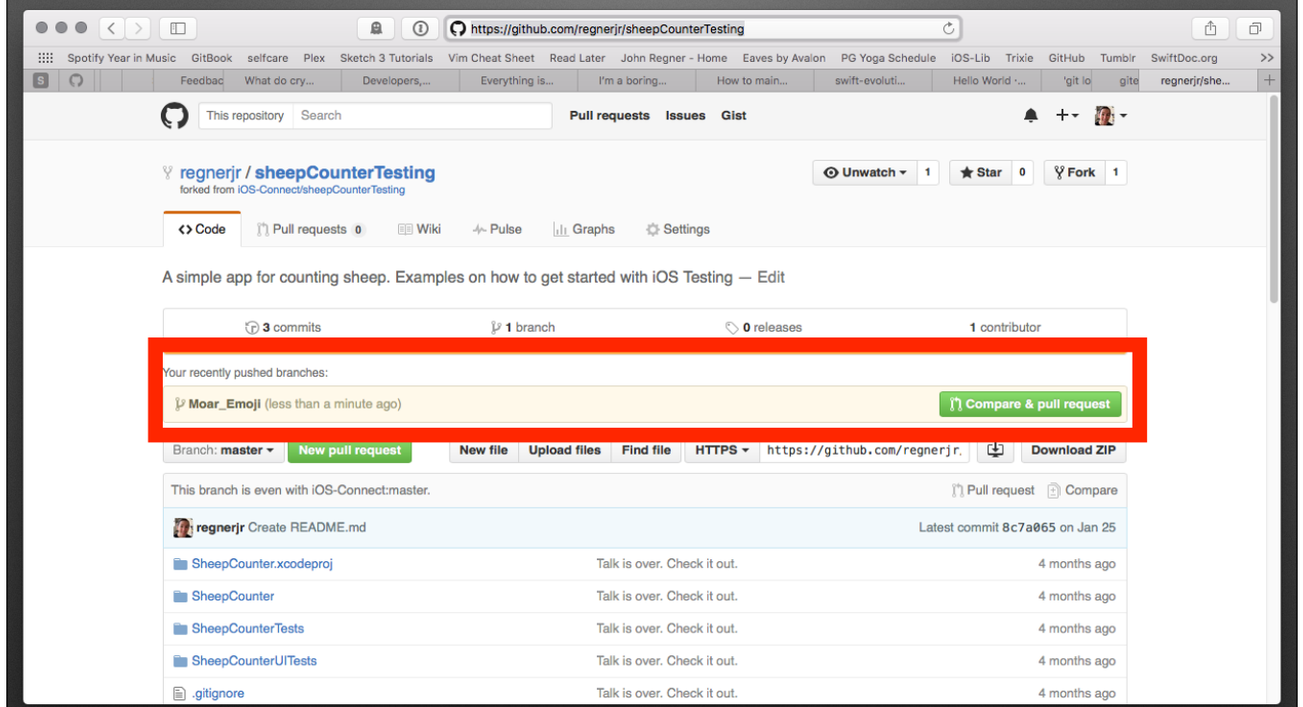
`git pull` will grab the changes from upstream and merge them with the current branch.
You'll probably use this for master to just pull down your teams changes.
Then maybe Rebase your own work on top.

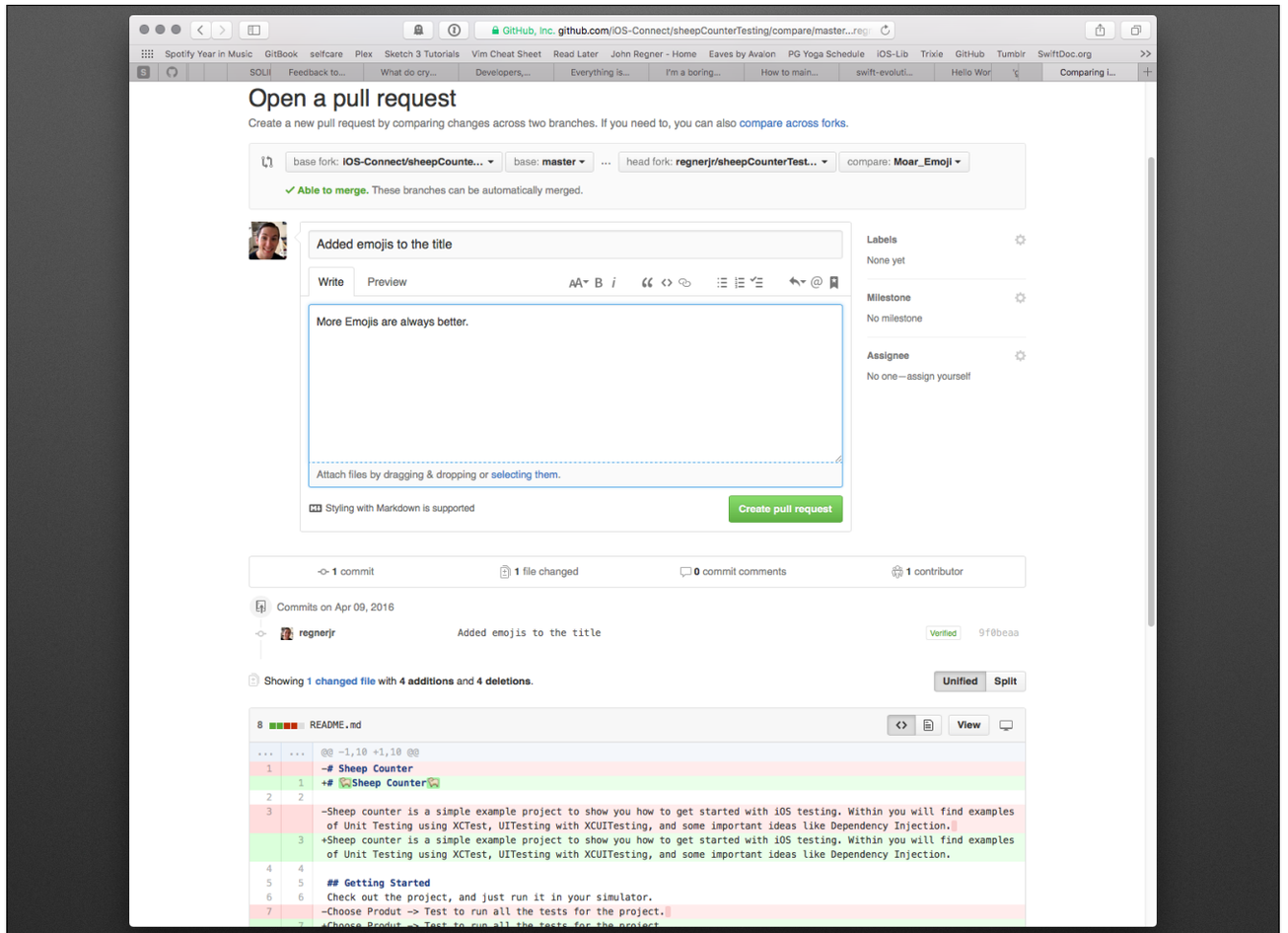
Pull Request!


conormaynard-germany

How to Pull Request (PR)

- Fork - Github copy
- Clone Your new fork - bring your copy to your machine
- Branch - With epic name
- Commits - to fix bugs and add features
- Push - Put your changes on Github (your fork)
- Propose a PR - Ask the other fork to include your code


https://github.com/regnerjr/sheepCounterTesting
This repository
Search
Pull requests
Issues
Gist
regnerjr / sheepCounterTesting
forked from iOS-Connect/sheepCounterTesting
Unwatch 1
Star 0
Fork 1
Code
Pull requests 0
Wiki
Pulse
Graphs
Settings
A simple app for counting sheep. Examples on how to get started with iOS Testing — Edit
3 commits
1 branch
0 releases
1 contributor
Your recently pushed branches:
Moar_Emoji (less than a minute ago)
Compare & pull request
Branch: master
New pull request
New file
Upload files
Find file
HTTPS
https://github.com/regnerjr.
Download ZIP
This branch is even with iOS-Connect:master.
Pull request
Compare
regnerjr Create README.md
Latest commit 8c7a065 on Jan 25
SheepCounter.xcodeproj
Talk is over. Check it out.
4 months ago
SheepCounter
Talk is over. Check it out.
4 months ago
SheepCounterTests
Talk is over. Check it out.
4 months ago
SheepCounterUITests
Talk is over. Check it out.
4 months ago
.gitignore
Talk is over. Check it out.
4 months ago


GitHub, Inc. github.com/iOS-Connect/sheepCounterTesting/compare/master...reg
Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.
base fork: iOS-Connect/sheepCounte... base: master ... head fork: regnerjr/sheepCounterTest... compare: Moar_Emoji
Able to merge. These branches can be automatically merged.
Added emojis to the title
Write
Preview
More Emojis are always better.
Attach files by dragging & dropping or selecting them.
Styling with Markdown is supported
Create pull request
Labels
None yet
Milestone
No milestone
Assignee
No one—assign yourself
1 commit
1 file changed
0 commit comments
1 contributor
Commits on Apr 09, 2016
regnerjr
Added emojis to the title
Verified
9f0beaa
Showing 1 changed file with 4 additions and 4 deletions.
Unified
Split
8
README.md
View
@@ -1,10 +1,10 @@
-# Sheep Counter
+# Sheep Counter
-Sheep counter is a simple example project to show you how to get started with iOS testing. Within you will find examples of Unit Testing using XCTest, UITesting with XCUITesting, and some important ideas like Dependency Injection.
+Sheep counter is a simple example project to show you how to get started with iOS testing. Within you will find examples of Unit Testing using XCTest, UITesting with XCUITesting, and some important ideas like Dependency Injection.
Getting Started
Check out the project, and just run it in your simulator.
-Choose Produt -> Test to run all the tests for the project.

Questions?

John Regner

@john_regner github.com/regnerjr

Resources

- http://guides.github.com
- http://www.learnenough.com/git-tutorial
- http://gitready.com
- http://git-scm.com

Grab Bag Section

Other awesome commands

- `git bisect` Performs binary search to find where a bug was introduced
- `git cherry-pick` Move a single commit from one branch to another
- `git grep` search through your repo. Only looking in places that are tracked by git, not your build folder.

Demo Interactive rebase