TODAY'S GOALS

1. Install Git.
2. Get a Github Account
3. Download a Repo
4. Fork a Repo
5. Create a Branch
6. Create a Pull Request
7. Create your own repo

Today's Secondary Goals

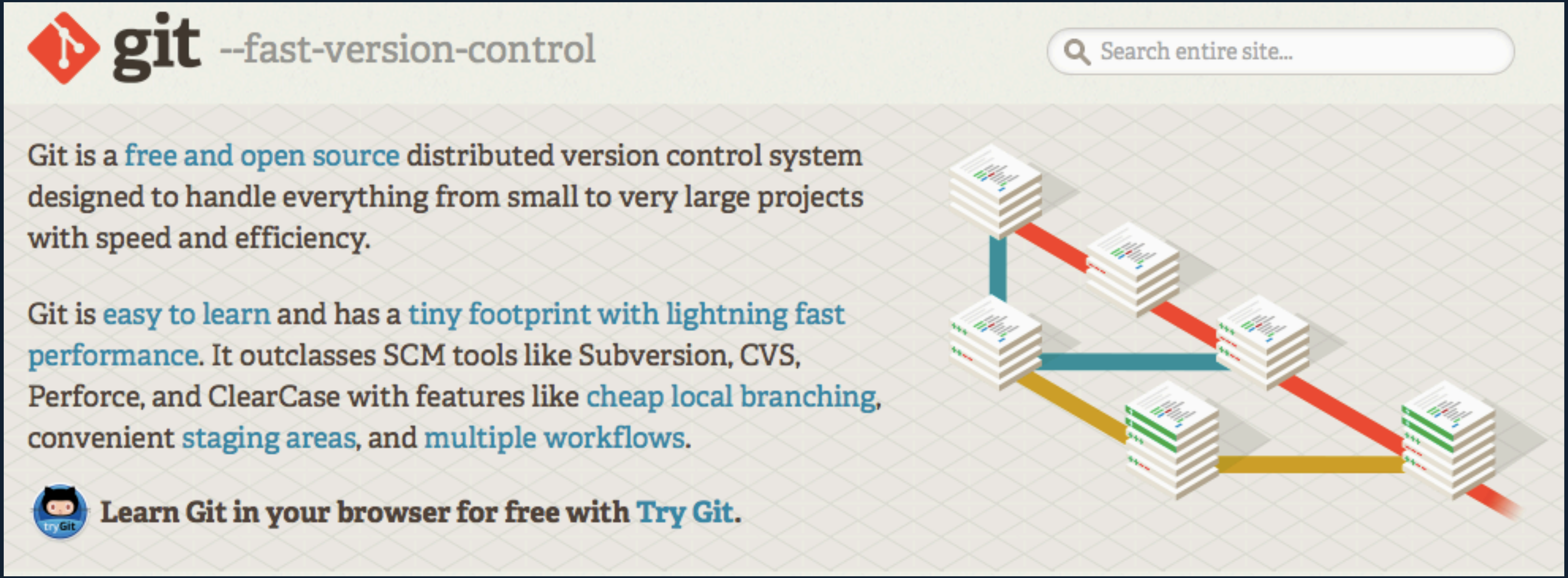
1. Know what Git is.
2. Know why you should care.

WHAT IS GIT?

A better way to download open source software.

```
git clone https://github.com/jquery/jquery.git
```

NO, WHAT IS GIT?



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

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

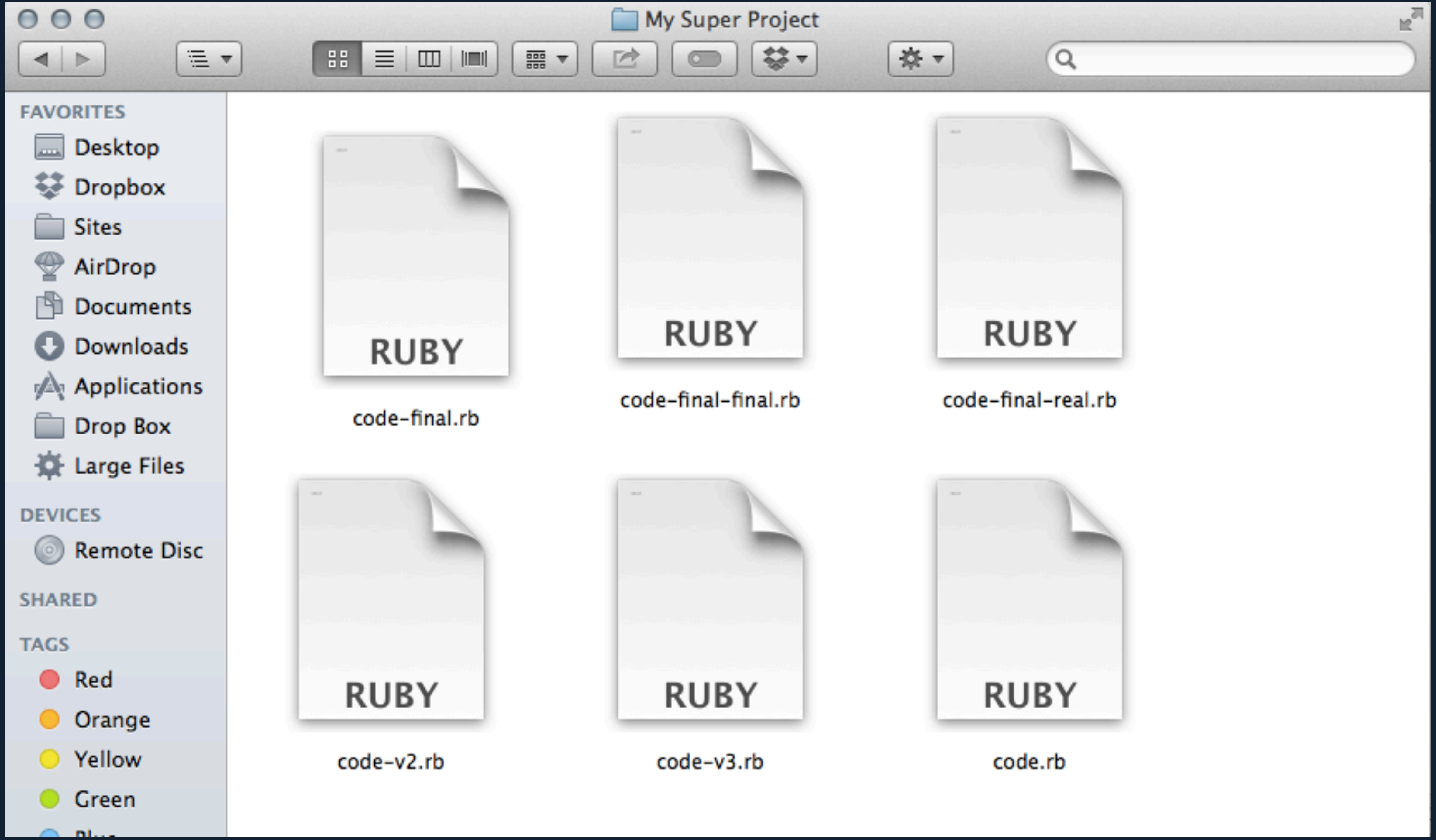
Learn Git in your browser for free with Try Git.

What is version control?

What is version control?

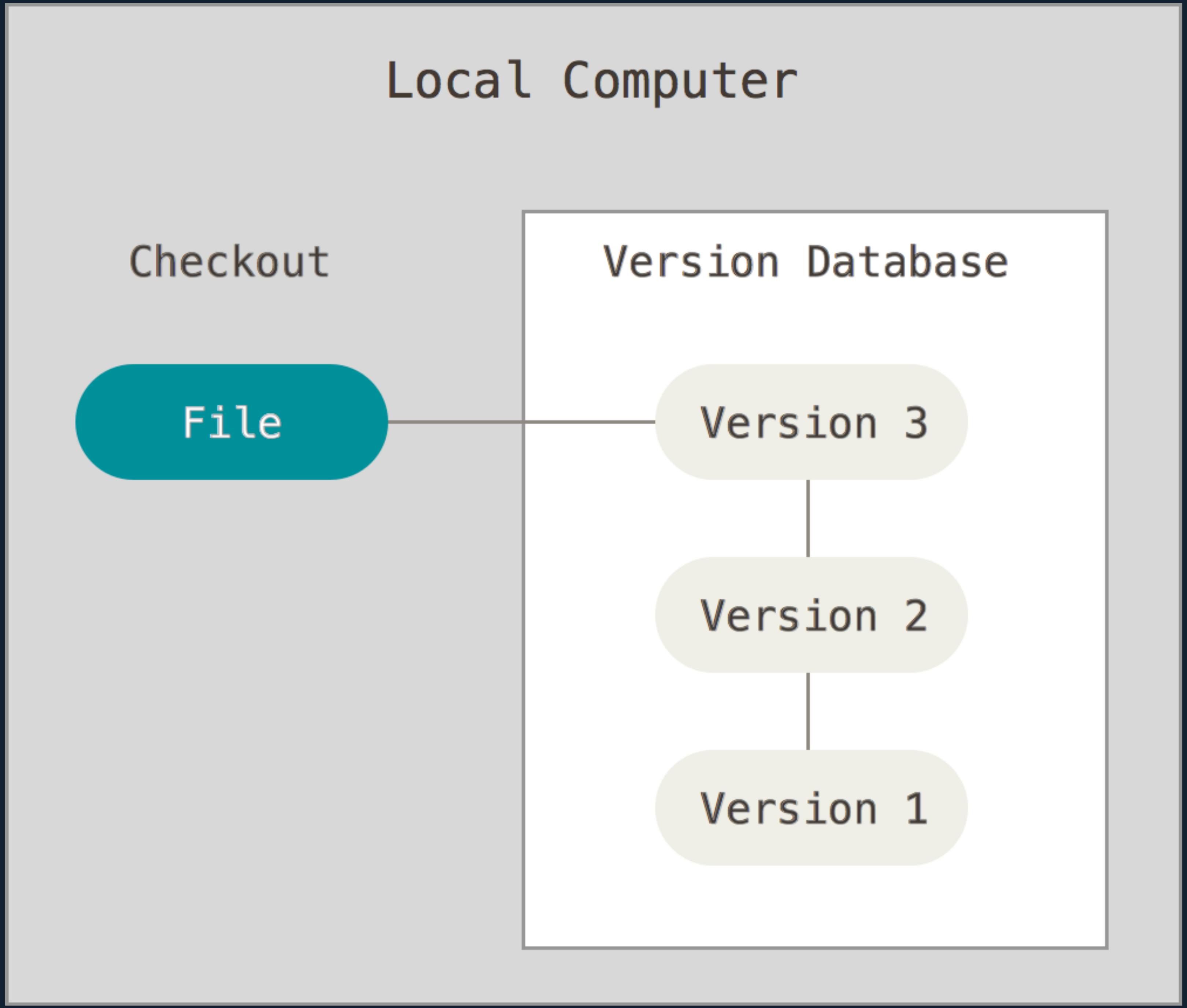
UNDO

VERSIONED FILES



code-final.rb
code-final-final.rb
code-final-real.rb
code-v2.rb
code-v3.rb
code.rb

LOCAL VERSION CONTROL



Local Computer

Checkout

File

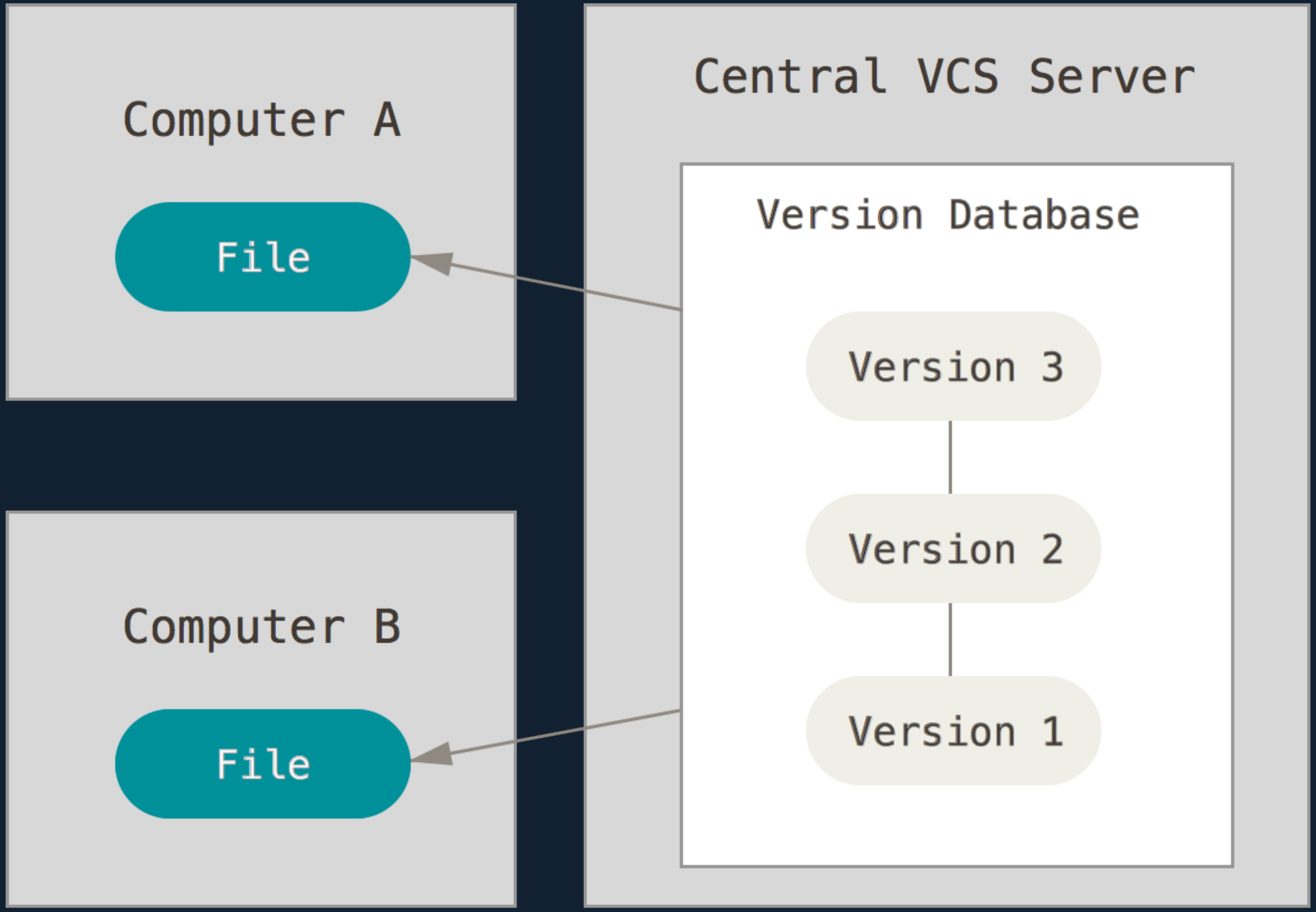
Version Database

Version 3

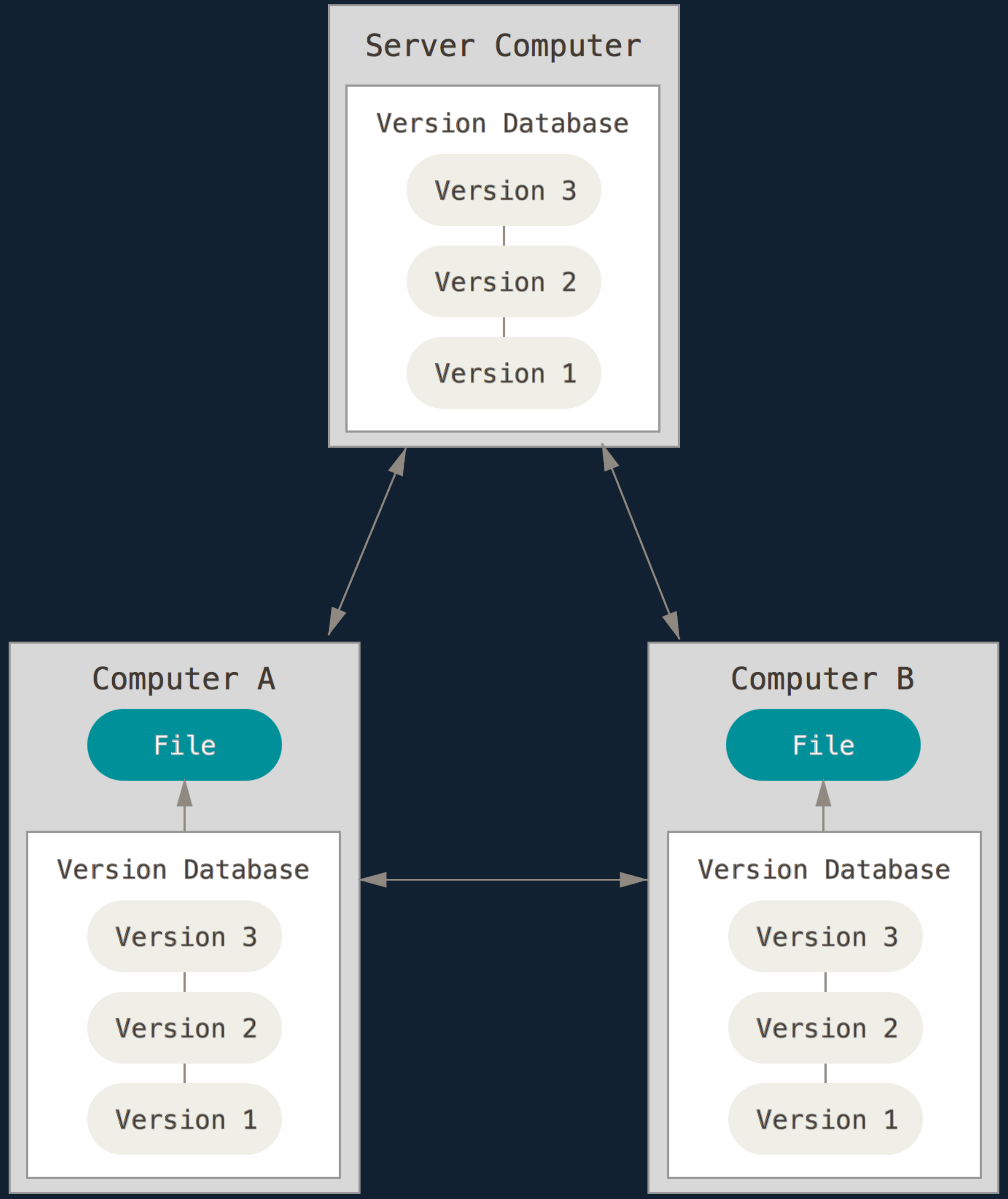
Version 2

Version 1

CENTRALIZED VERSION CONTROL



Distributed Version Control

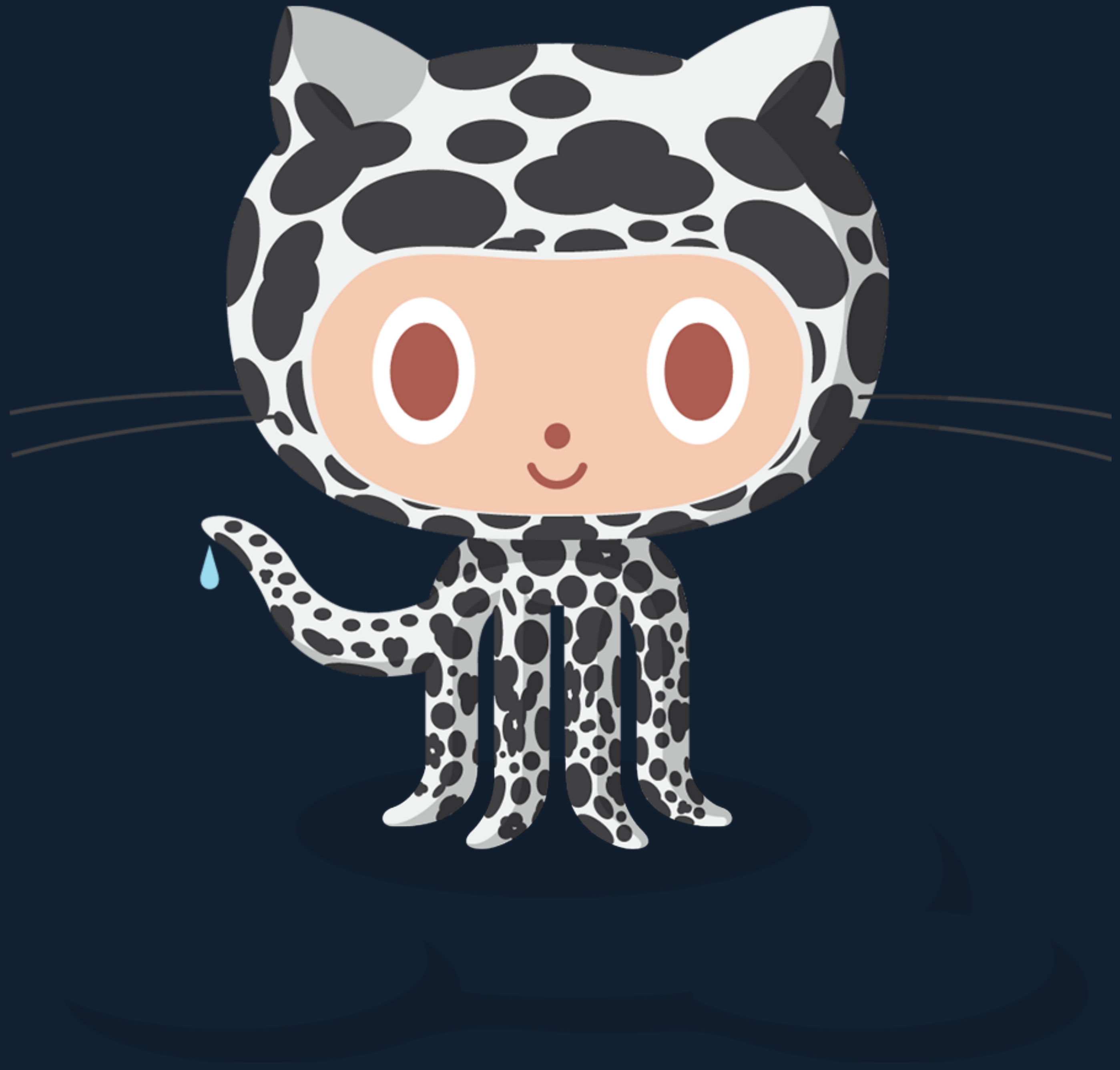


SO, WHAT IS GIT?

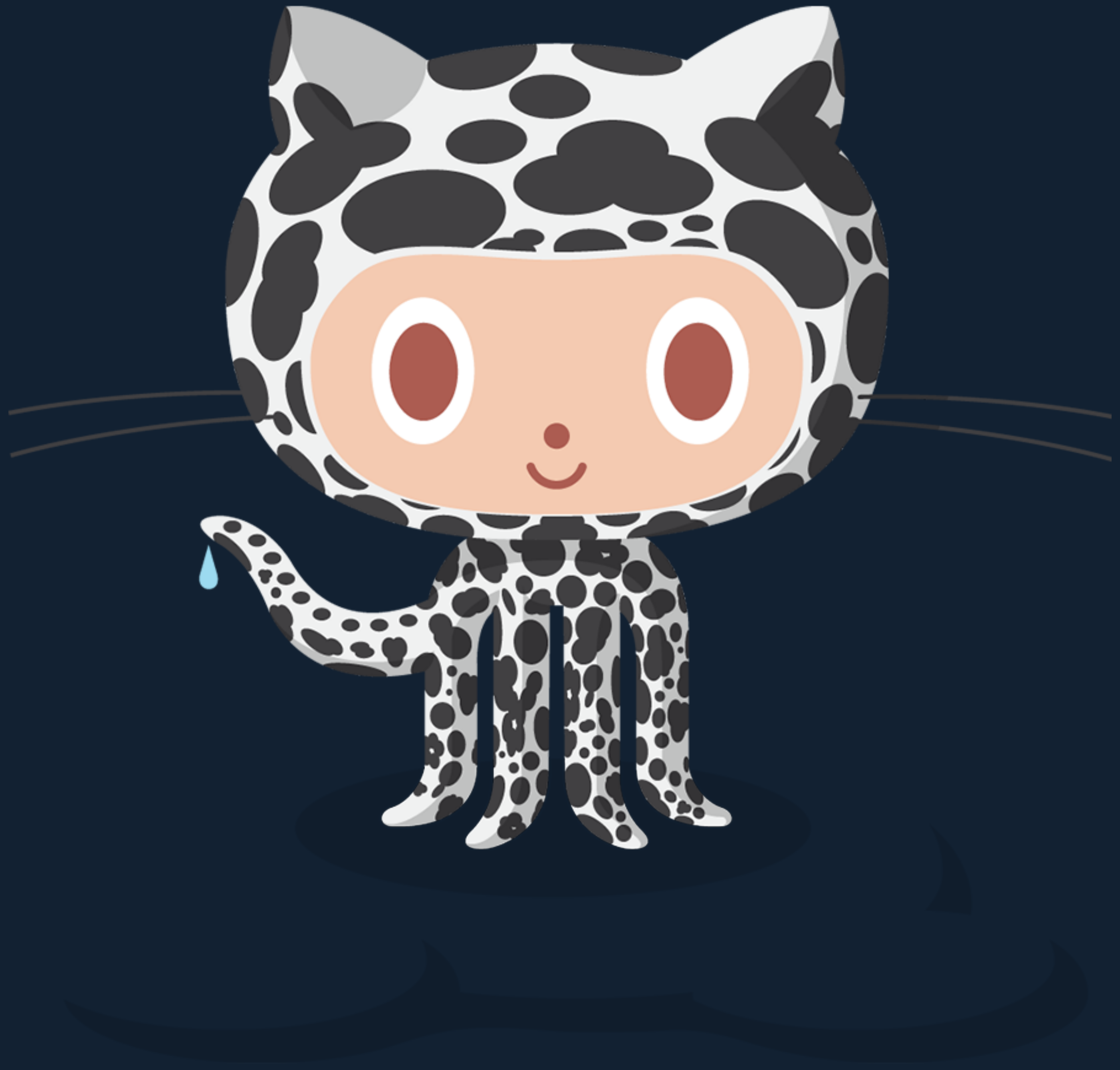
GIT IS A WAY TO **MANAGE** AND **COLLABORATE** ON LARGE PROJECTS.

WHAT IS GITHUB?

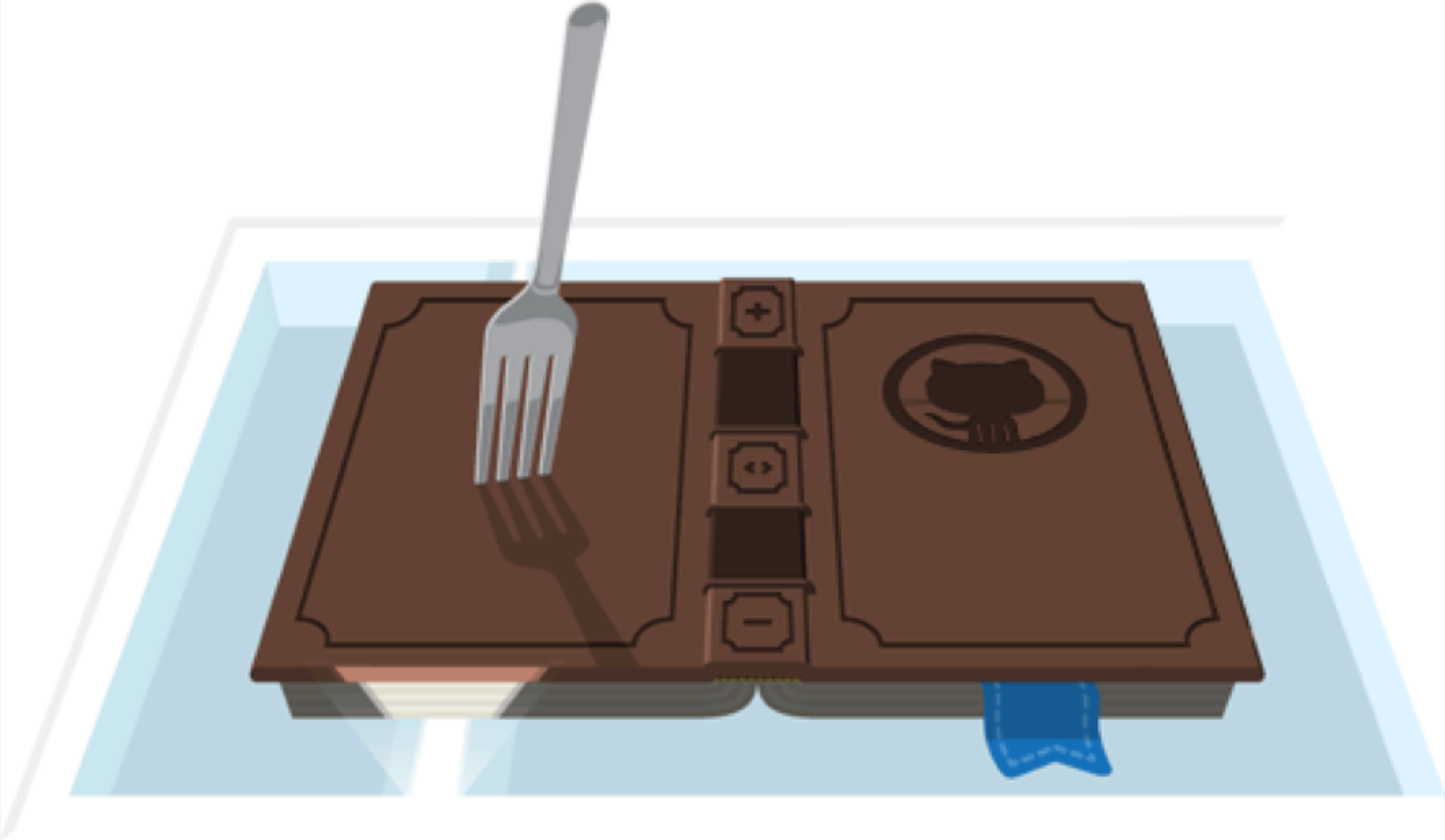
A PLACE FOR CODE.



A PLACE FOR PUBLIC CODE.



A PLACE FOR FORKS.



USE SOMEONE ELSE'S

PROJECT

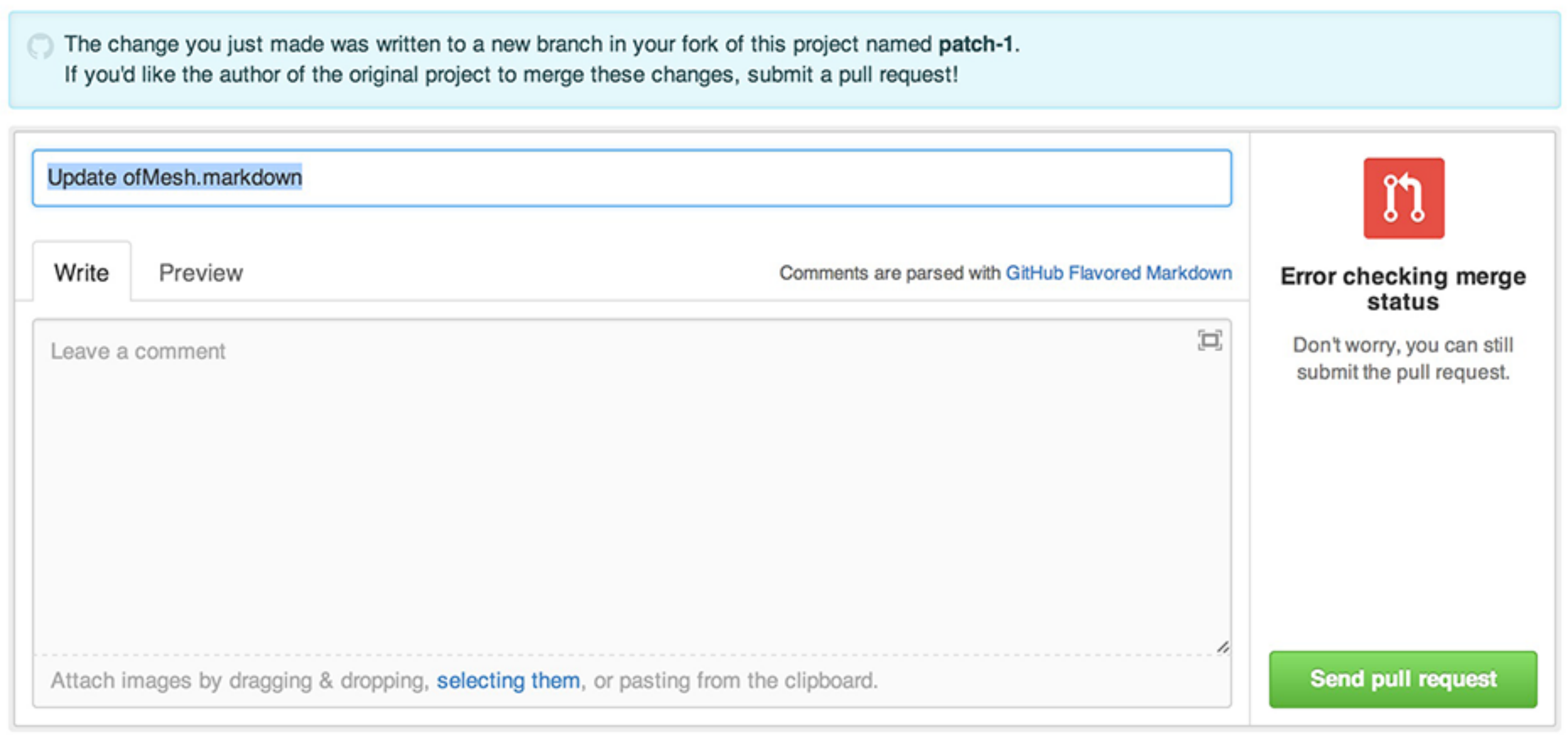
AS A

STARTING

POINT

FOR YOUR OWN IDEA.

PULL REQUESTS



The change you just made was written to a new branch in your fork of this project named **patch-1**.
If you'd like the author of the original project to merge these changes, submit a pull request!

Update ofMesh.markdown

Comments are parsed with GitHub Flavored Markdown

Leave a comment

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

Error checking merge status

Don't worry, you can still submit the pull request.

GitHub Flow:

Fork it. Clone it.
Branch it. Change it.
Add it. Commit it.
Push it. Pull Request it.

Three Trees:

1. working directory
   - This is the current directory.
2. staging/index
   - this is everything added, but not committed
3. HEAD
   - this is your last commit.

Three repos:

1. local
   - This is on your computer
2. origin
   - this is yours on github
3. upstream
   - this is the original, on github

Get Git

Command line:

`http://git-scm.com/downloads`

GUI:

`https://windows.github.com`
or
`https://mac.github.com`

(make it so)

YOUR FIRST DEFENSE

```
git help <anything>
```

ADD YOUR USER INFO

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com

# make it pretty
git config --global color.ui true
```

Add your editor

```
git config --global core.editor subl
# or
git config --global core.editor mate
# or
git config --global core.editor nano
# if you're crazy
git config --global core.editor vim
# if you're crazier
git config --global core.editor emacs
```

CREATE SSH-KEYS

Follow these instructions:
https://help.github.com/articles/generating-ssh-keys/

TL;DR.

```
ssh-keygen -t rsa -C "your_email@example.com"
pbcopy < ~/.ssh/id_rsa.pub
# Upload to github
ssh -T git@github.com
```

COMMANDS

git init

CREATE A NEW REPOSITORY

```
git init pandatracker
cd pandatracker

# or, if you already have a directory:
git init .
```

git status

WHAT'S GOING ON?

```
git status
```

git status

WHAT'S GOING ON?

```
On branch master
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:   git.txt

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

git clone

DOWNLOAD A REMOTE REPOSITORY

```
git clone https://github.com/tadejm/sing-along.git
```

git clone

DOWNLOAD A REMOTE REPOSITORY

```
Cloning into 'sing-along'...
remote: Counting objects: 89, done.
remote: Total 89 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (89/89), done.
Checking connectivity... done.
```

git add <filename>

ADD A FILE TO STAGING

```
# add a single file
git add petunia.rb

# add the current directory
git add .
```

git commit -m "message"

COMMIT YOUR CHANGES TO HEAD

```
git commit -m "Added the interface to the petunia neurons"
```

git commit -m "message"

COMMIT YOUR CHANGES TO HEAD

```
[master 668ce5f] Added the interface to the petunia neurons
 1 file changed, 1 insertion(+)
```

git commit

Open your favorite text editor.

Meaningful commit messages

~~`wip`~~

Meaningful commit messages

~~`wip`~~

~~`bugfixes`~~

Meaningful commit messages

~~`wip`~~

~~`bugfixes`~~

~~`changed some stuff`~~

Meaningful commit messages

~~wip~~

~~bugfixes~~

~~changed some stuff~~

~~why on earth am I working at three am????~~

MEANINGFUL COMMIT MESSAGES

Added the interface to the petunia neurons

We were having issues with interfacing telepathically with the flowers growing in the garden, so we've added a system that allows us to amplify their nural messages using a combination of jQuery and Miracle Grow.

git push origin master

UPLOAD YOUR CODE

```
git push origin master
```

git push origin master

UPLOAD YOUR CODE

```
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

git push origin master

UPLOAD YOUR CODE

```
# if you don't have a remote
git remote add origin <server>

# then
git push origin master
```

git checkout -b <name>

CREATE A BRANCH

```
# create a branch
git checkout -b lasereyes

# go back to master
git checkout master

# push to a remote
git push origin lasereyes

# Delete the branch
git branch -d lasereyes
```

git pull

Get changes from a remote

```
git pull
```

MERGE CONFLICTS

Merge Conflicts

```
# Find out what's up
git status

# If you want your version of a file
git checkout --ours petunia.rb

# If you want their version of a file
git checkout -theirs rose.rb

# If you fixed it manually
git add daffoldil.rb
```

POWER TOOLS

git log

SEE THE HISTORY OF WHAT YOU'VE DONE

```
git log

commit 668ce5fb3739e13beb61d01077421f32499e86f6
Author: David Newbury <david.newbury@gmail.com>
Date:   Thu Jan 15 07:04:30 2015 -0500

    Added the interface to the petunia neurons

commit 8397296d7867ffd05d836a8c11a203dd15cc4652
Author: David Newbury <david.newbury@gmail.com>
Date:   Thu Jan 15 06:12:30 2015 -0500

    I really really made a thing
```

git stash

PULLING WHEN YOU HAVE UNCOMMITTED CHANGES

```
git stash
git pull upstream develop
git stash pop
```

git commit --amend

YOU FORGOT SOMETHING.

```
git commit -m "Fixed the wumpus"
# (discover the wumpus is still broken)
# ...
# (actually fix the wumpus)
git commit --amend
```

git checkout -- <filename>

UNDO YOUR CURRENT WORK

```
git checkout -- oops.rb
# oops.rb is now reverted at the last commit

git reset --hard
# All changes since your last commit are **GONE**.

git fetch origin
git reset --hard origin/master
# Everything is reverted to the master repo
```

.gitignore

Keep files out of the repo

```
_site
*~
*.pyc
.DS_Store
/tmp
```

Note: This is a file, not a command.

git diff

SEE THE CHANGES

```
diff --git a/petunia.rb b/petunia.rb
index 8895793..e6021aa 100644
--- a/petunia.rb
+++ b/petunia.rb
@@ -1,4 +1,4 @@
 # Do it.
 invigorate.the.petunias(100)

-run.from(petunias)
+run.from(petunias).faster!
```

PROJECT TIME

HTTPS://GITHUB.COM/WORKERGNOME/LIES

1. Fork it.
2. Pull it locally
3. Create a branch.
4. Add a new file (yourlastname.txt)
5. Add and commit your changes
6. Push your branch to github
7. Submit a pull request.

HELPFUL LINKS

HTTP://ROGERDUDLER.GITHUB.IO/GIT-GUIDE/
HTTP://ROGERDUDLER.GITHUB.IO/GIT-GUIDE/FILES/GIT_CHEAT_SHEET.PDF

...stack overflow.