GIT / GITHUB

WHAT IS (G)IT?

- CODE VERSION CONTROL
- 28M DEVELOPERS
- 85M PROJECTS (REPOSITORIES)

A TOOL YOU (DEVELOPER) WILL BE USING TO HOST YOUR DS PROJECTS (REPOSITORIES) IN ORDER TO COLLABORATE WITH ONE ANOTHER AND MAINTAIN VERSION HISTORY

AWESOME!
HOW DO I USE IT?

STEP 1

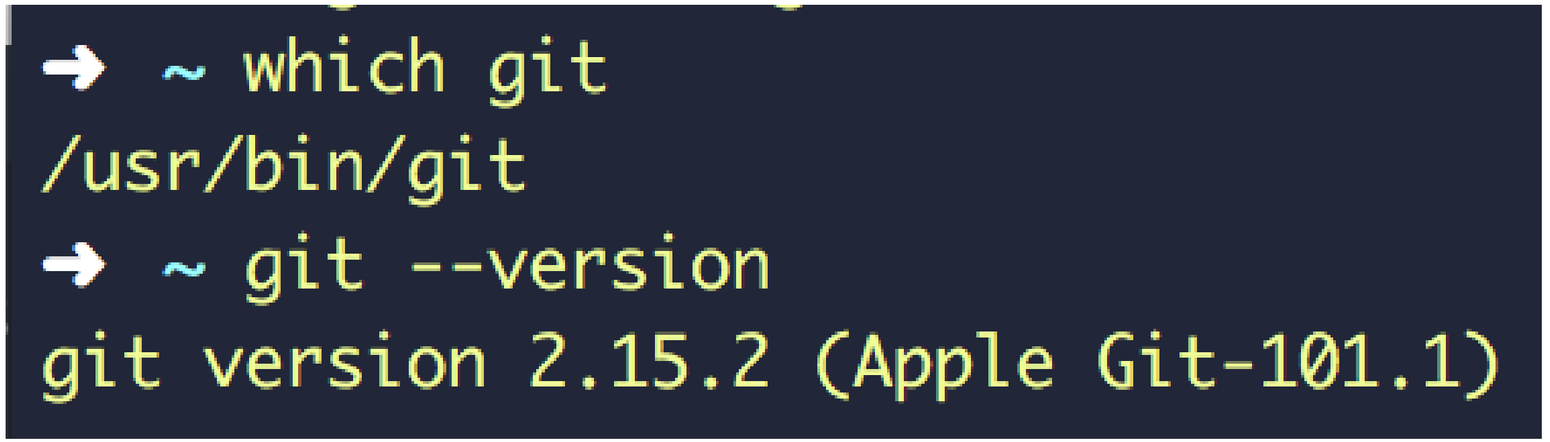
INSTALLING GIT

- LINUX
- WINDOWS
- MAC

STEP 2

MAKING SURE ITS INSTALLED..

OPEN YOUR TERMINAL



STEP 3

CREATE A PROJECT!

WITH THE TERMINAL STILL OPEN..

```
➜  ~ mkdir git-test-project
➜  ~ cd git-test-project
➜  git-test-project git init
Initialized empty Git repository in /Users/taylorperkins/git-test-project/.git/
➜  git-test-project git:(master) ls -a
.    ..    .git
```

WHAT DOES IT MEAN?

1. create a new directory called `git-test-project`
2. enter that folder
3. initialize a new `git` project
4. show me the contents inside current directory

STEP 4
CREATE A FILE

```
➜  git-test-project git:(master) ls
➜  git-test-project git:(master) echo "writing with master" >> project.txt
➜  git-test-project git:(master) ✗ ls
project.txt
➜  git-test-project git:(master) ✗ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        project.txt

nothing added to commit but untracked files present (use "git add" to track)
```

WHAT DOES IT MEAN?

1. create a new file with `writing with master` as the content
2. list files with `ls`
3. show current status of my git project in master

STEP 5

ADD AND COMMIT CHANGES

```
➜  git-test-project git:(master) ✗ git add .
➜  git-test-project git:(master) ✗ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   project.txt

➜  git-test-project git:(master) ✗ git commit -am "Created new file on master."
[master (root-commit) b213739] Created new file on master.
 1 file changed, 1 insertion(+)
 create mode 100644 project.txt
➜  git-test-project git:(master)
```

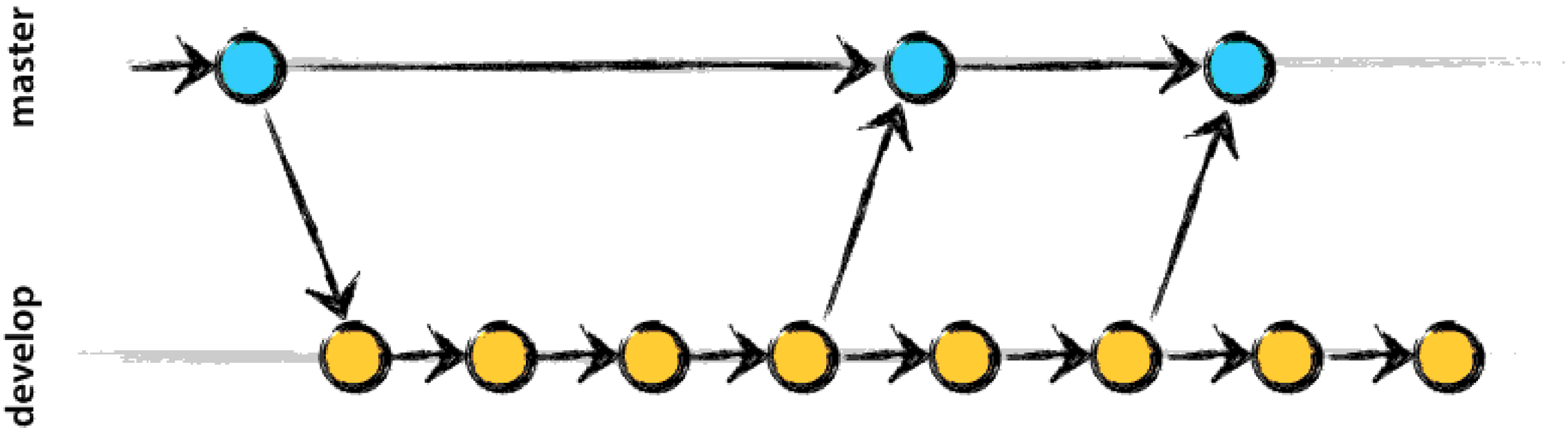
WHAT DOES IT MEAN?

1. add all modified and new (untracked) files in the current directory and all subdirectories to the staging area (a.k.a. the index)
2. show status
3. record changes to the repository (save)

MASTER?

The beginning of your source control!

Let's talk branches..



master == production (final)

develop == features, ideas, tests, exploratory

STEP 6

CREATING DEVELOP

```
➜  git-test-project git:(master) git checkout -b develop
Switched to a new branch 'develop'
➜  git-test-project git:(develop) git branch
* develop
  master
```

1. create and move to new branch, `develop`
2. list all branches

```
➜ git-test-project git:(develop) cat project.txt
writing with master
➜ git-test-project git:(develop) echo "writing with develop" >> project.txt
➜ git-test-project git:(develop) ✗ cat project.txt
writing with master
writing with develop
➜ git-test-project git:(develop) ✗ git diff | cat
diff --git a/project.txt b/project.txt
index 49b596e..fb59252 100644
--- a/project.txt
+++ b/project.txt
@@ -1 +1,2 @@
 writing with master
+writing with develop
➜ git-test-project git:(develop) ✗ git add .
➜ git-test-project git:(develop) ✗ git commit -am "Added in a line for develop"
[develop ac1d728] Added in a line for develop
 1 file changed, 1 insertion(+)
➜ git-test-project git:(develop) git checkout master
Switched to branch 'master'
➜ git-test-project git:(master) cat project.txt
writing with master
➜ git-test-project git:(master)
```

STEP 7

MAKE CHANGE WITH DEVELOP

WHAT DOES IT MEAN?

1. edit project.txt with develop branch
2. show diff in changes between master and develop
3. add and commit changes (same as before..)
4. change branch to master
5. assert no diff

STEP 8

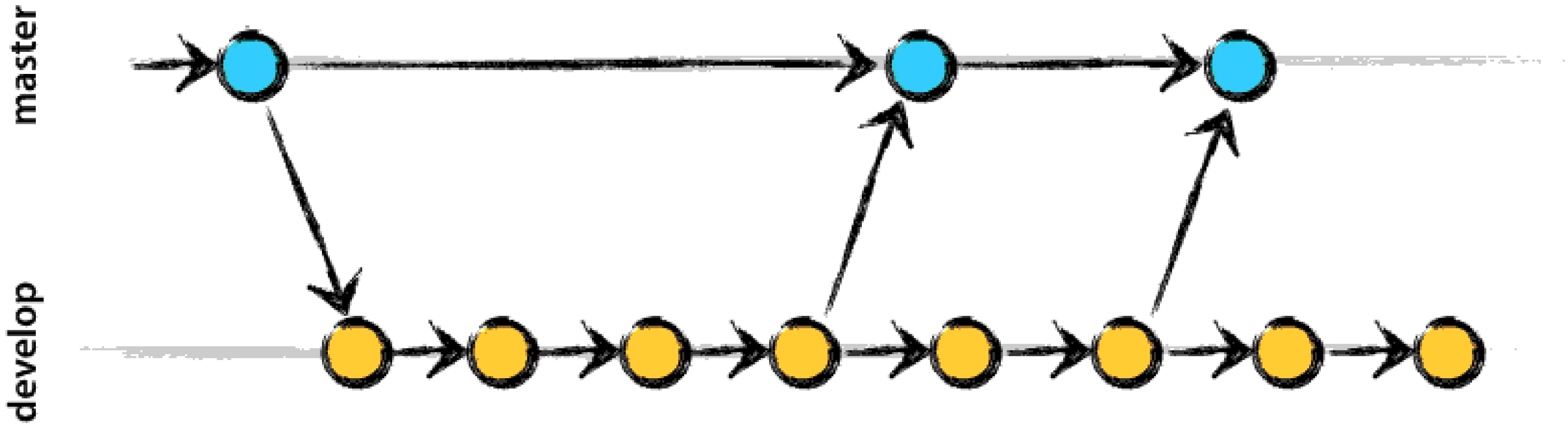
MERGE DEVELOP CHANGES INTO MASTER

```
➜  git-test-project git:(master) git merge develop
Updating 53dbc00..91b3dec
Fast-forward
 project.txt | 1 +
 1 file changed, 1 insertion(+)
➜  git-test-project git:(master) cat project.txt
writing with master
writing with develop
➜  git-test-project git:(master) git branch -d develop
Deleted branch develop (was 91b3dec).
➜  git-test-project git:(master) git branch
* master
➜  git-test-project git:(master)
```

WHAT DOES IT MEAN?

1. merge changes made in develop into master
2. assert the updates
3. delete branch develop

GIT LIFECYCLE, COMPLETE



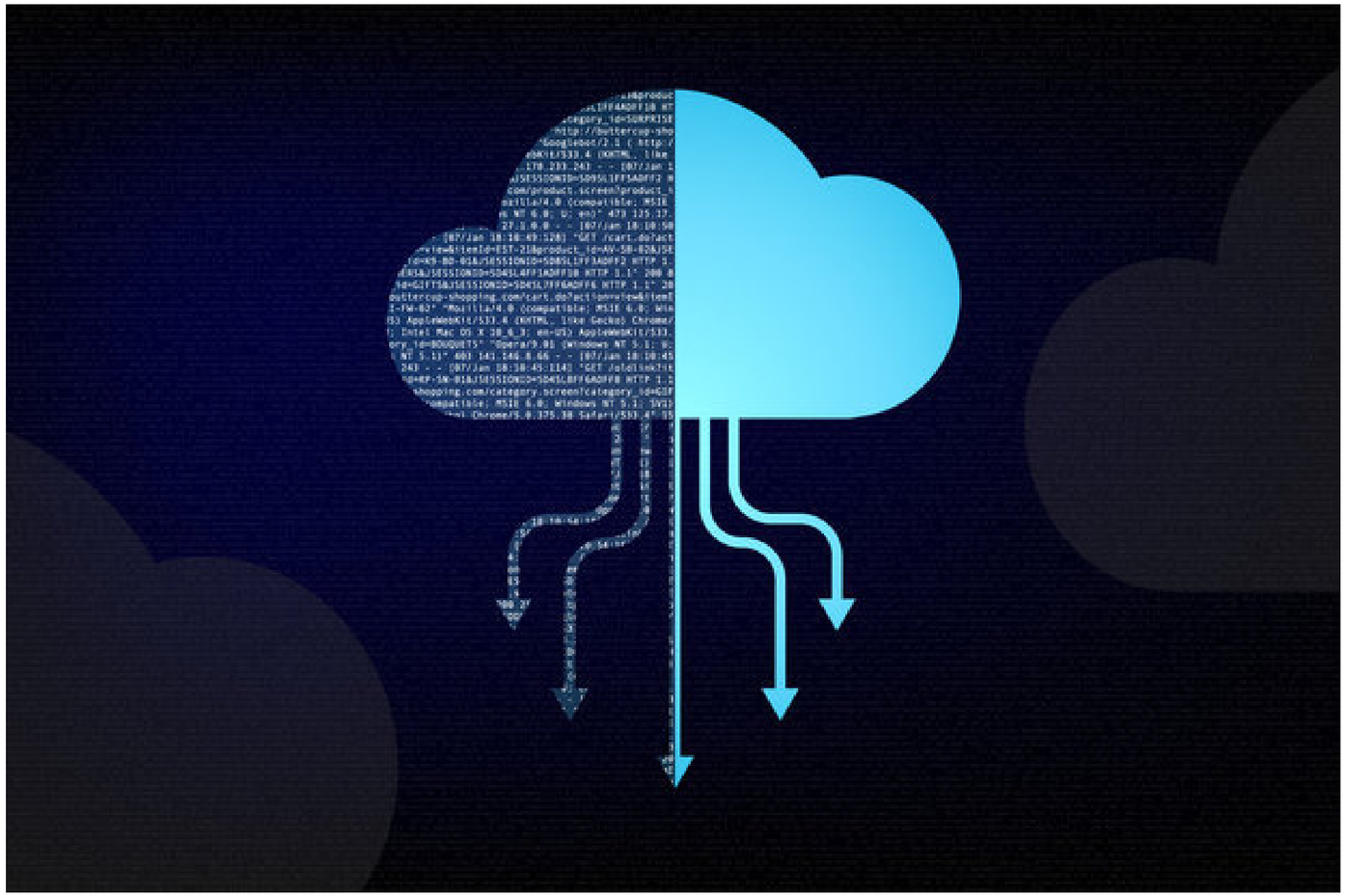
RINSE AND REPEAT

REVIEW!

git init	- initialize new git project
git status	- show status of files ready to be committed, or in staging
git add <files>	- add changes to be committed
git commit -am <message>	- "save" changes
git diff	- show changes between branches
git branch	- list branches
git checkout <branch>	- switch to existing branch
git checkout -b <branch>	- create new branch, and switch
git branch -d <branch>	- delete branch
git merge <branch>	- merge <branch> into current branch

GITHUB

CODE WITH EACH OTHER



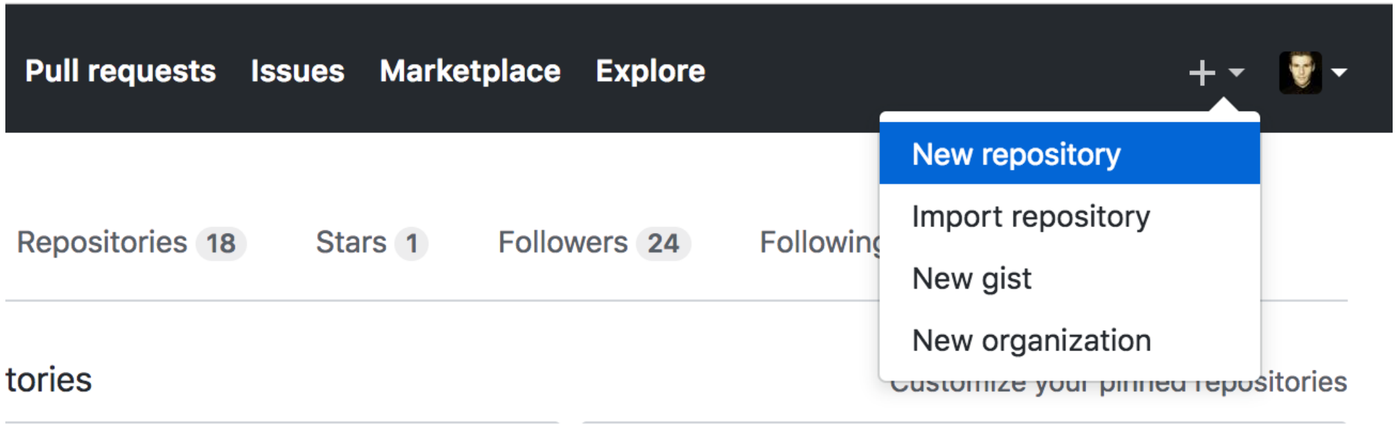
STEP 1

CREATE GITHUB ACCOUNT

EMAIL, PASSWORD.

SETUP SSH

STEP 2
CREATE GITHUB PROJECT



1. assign name
2. put in a description
3. click `Create Repository`
4. copy `…or push an existing repository from the command line` code

STEP 3

PUSH PROJECT FROM GIT TO GITHUB

```
➜  git-test-project git:(master) git remote add origin git@github.com:taylorperkins/git-test-project.git
fatal: remote origin already exists.
➜  git-test-project git:(master) git push -u origin master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 504 bytes | 504.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:taylorperkins/git-test-project.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
➜  git-test-project git:(master)
```

taylorperkins / git-test-project

Watch 0 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

test github project

Edit

Add topics

2 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

taylorperkins Added in a line for develop — Latest commit 91b3dec an hour ago

project.txt	Added in a line for develop	an hour ago

Help people interested in this repository understand your project by adding a README. Add a README

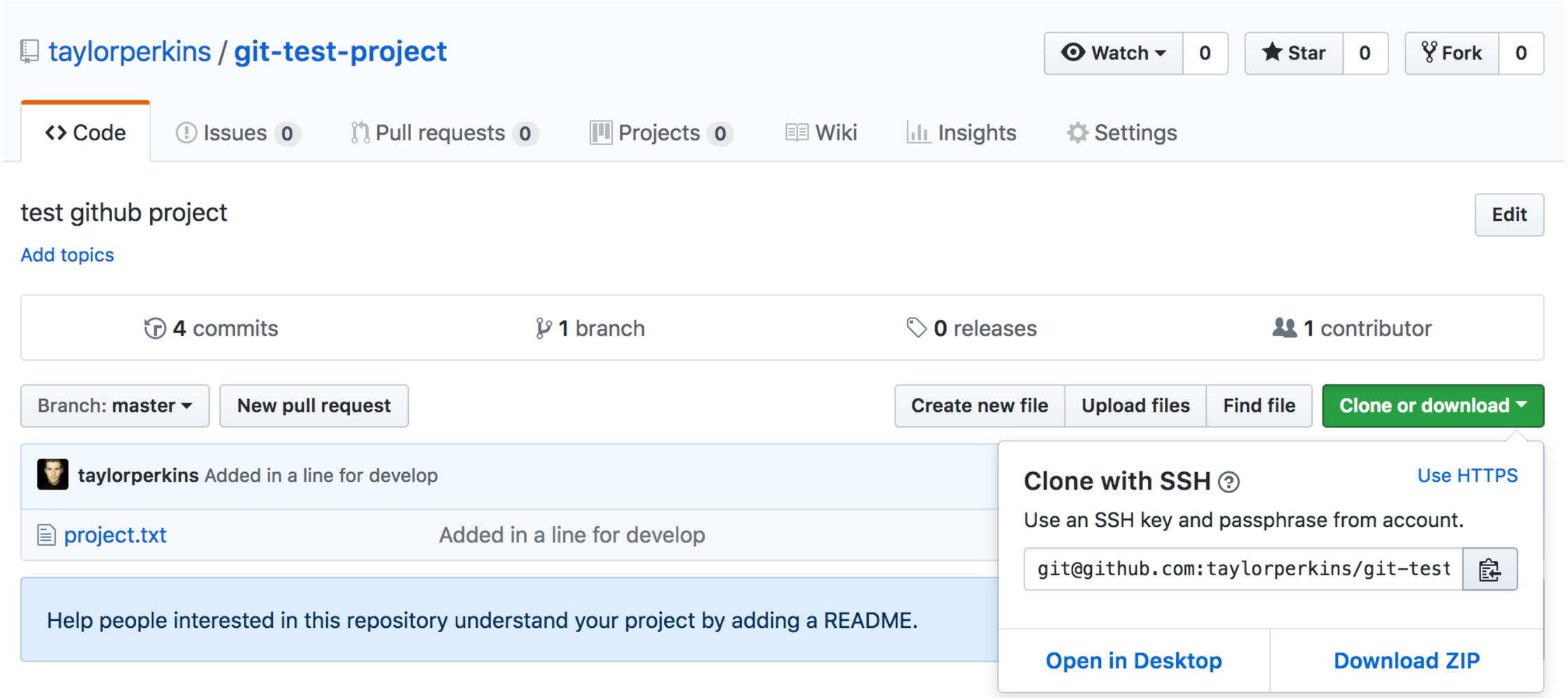
CONNECTION COMPLETE!

You have successfully started a project from your command line (terminal), and pushed up all of your code to Github to be seen by, shared, and contributed to.

Great job!

STEP 4 (CONTRIBUTION)

CLONE DOWN REPOSITORY



STEP 4 (CONTINUED)

CLONE DOWN REPOSITORY

```
➜  Desktop cd projects
➜  projects git clone git@github.com:taylorperkins/git-test-project.git
Cloning into 'git-test-project'...
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 8 (delta 2), reused 8 (delta 2), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (2/2), done.
➜  projects cd git-test-project
➜  git-test-project git:(master) cat project.txt
writing with master
writing with develop
➜  git-test-project git:(master)
```

WHAT DOES IT MEAN?

1. copy the cloning url to your clipboard
2. clone down the project you want to work on. You are not starting this project, but contributing to someone else's project.
3. assert the project exists

STEP 6

ADD CONTRIBUTION

```
➜ git-test-project git:(master) git checkout -b taylor/contribute-to-project-file
Switched to a new branch 'taylor/contribute-to-project-file'
➜ git-test-project git:(taylor/contribute-to-project-file) echo "Taylor Perkins contributing to project." >> project.txt
➜ git-test-project git:(taylor/contribute-to-project-file) ✗ git status
On branch taylor/contribute-to-project-file
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:   project.txt

no changes added to commit (use "git add" and/or "git commit -a")
➜ git-test-project git:(taylor/contribute-to-project-file) ✗ git diff | cat
diff --git a/project.txt b/project.txt
index fb59252..afef62b 100644
--- a/project.txt
+++ b/project.txt
@@ -1,2 +1,3 @@
 writing with master
 writing with develop
+Taylor Perkins contributing to project.
➜ git-test-project git:(taylor/contribute-to-project-file) ✗ git add .
➜ git-test-project git:(taylor/contribute-to-project-file) ✗ git commit -am "Added personal edition to project."
[taylor/contribute-to-project-file 298f505] Added personal edition to project.
 1 file changed, 1 insertion(+)
➜ git-test-project git:(taylor/contribute-to-project-file) git push -u origin taylor/contribute-to-project-file
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 331 bytes | 331.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:taylorperkins/git-test-project.git
 * [new branch]      taylor/contribute-to-project-file -> taylor/contribute-to-project-file
Branch 'taylor/contribute-to-project-file' set up to track remote branch 'taylor/contribute-to-project-file' from 'origin'.
➜ git-test-project git:(taylor/contribute-to-project-file)
```

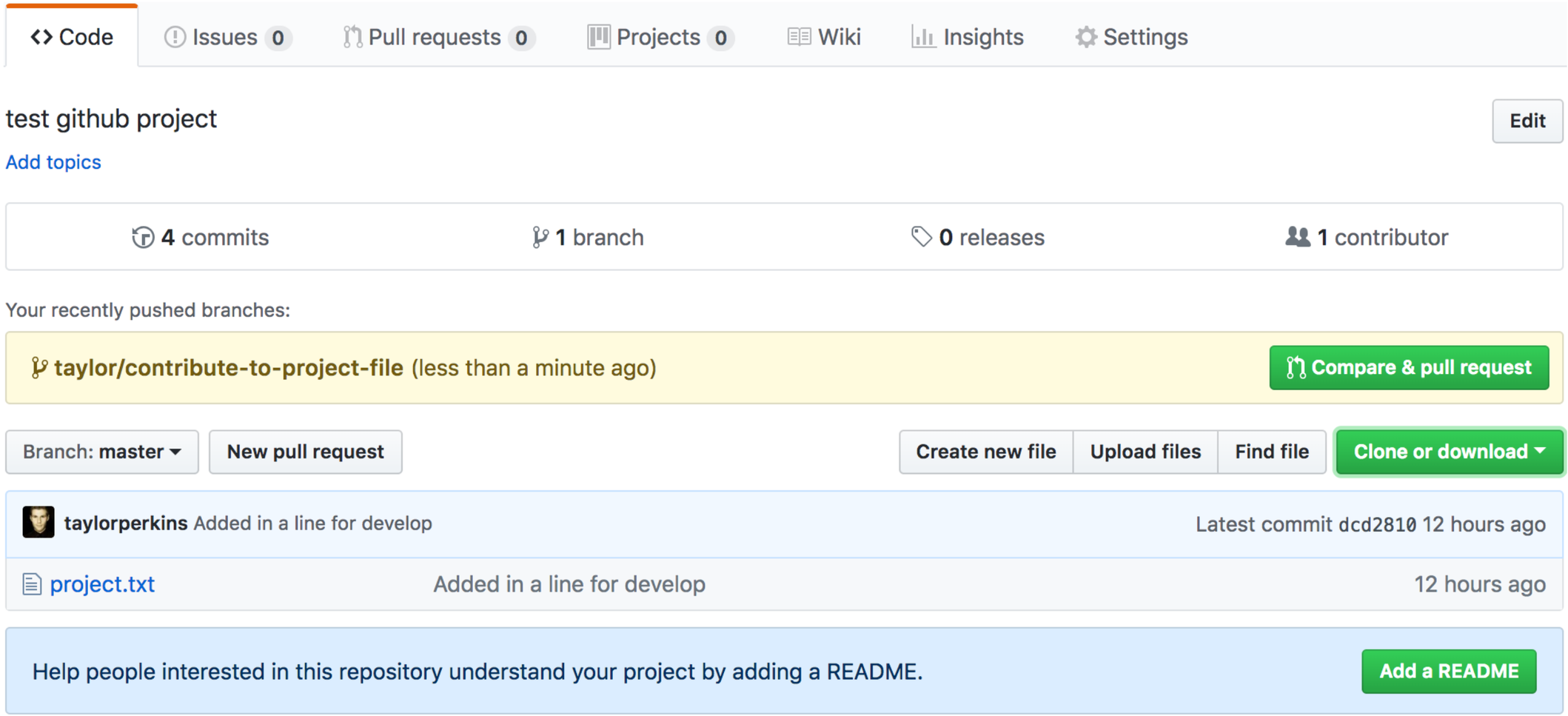
WHAT DOES IT MEAN?

It is everything we have been doing up to this point..

1. contribute to the project in some way
2. check the status and diff
3. add and commit the changes
4. push up your changes to Github

STEP 7

CREATE PULL REQUEST



STEP 7 (CONTINUED)

CREATE PULL REQUEST

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: master ← compare: taylor/contribute-to-project... ✓ Able to merge. These branches can be automatically merged.

Added personal edition to project.

Write | Preview

I added a line to the project.txt file

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

Styling with Markdown is supported

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

STEP 7 (CONTINUED)

CREATE PULL REQUEST

Added personal edition to project. #1

Edit

Open **taylorperkins** wants to merge 1 commit into `master` from `taylor/contribute-to-project-file`

Conversation 0 | Commits 1 | Checks 0 | Files changed 1 | +1 −0

taylorperkins commented just now — Owner

I added a line to the project.txt file

`Added personal edition to project.` `298f505`

Add more commits by pushing to the **`taylor/contribute-to-project-file`** branch on **taylorperkins/git-test-project**.

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications

WHAT DOES IT MEAN?

Put your code up for review!!

1. Create a "pull request" (request a code review) for your new changes against the branch you want to contribute to (master)
2. Wait for review by main authors or other contributors
3. merge it in!

STEP 8

PULL DOWN MERGED CHANGES

```
➜  git-test-project git:(taylor/contribute-to-project-file) git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
➜  git-test-project git:(master) git pull
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From github.com:taylorperkins/git-test-project
   dcd2810..e775244  master     -> origin/master
Updating dcd2810..e775244
Fast-forward
 project.txt | 1 +
 1 file changed, 1 insertion(+)
➜  git-test-project git:(master) cat project.txt
writing with master
writing with develop
Taylor Perkins contributing to project.
➜  git-test-project git:(master)
```

WHAT DOES IT MEAN?

Now that your code has been merged.. You are able to pull down a new version of master, and witness your contributions to "production" code.

1. Switch to master branch
2. Pull down the most recent version from Github
3. Assert updates!

LIFECYCLE COMPLETE!!

We have..

1. Cloned down a project
2. Made contributions
3. Pushed up changes
4. Requested review
5. Merged in new code
6. Pulled down new version into local git
7. Checked that the changes exist!

REVIEW!

git clone <url>	- pull down repository from github onto your machine
git remote add origin <url>	- link your local repository to a remote repository
git push -u origin <branch>	- push your branch to the origin url
git pull	- pull down most recent changes of current branch