

Git Help

An introduction to Git and GitHub

What is Git?

Why use Git?

Keeping track of code changes

- `git init`
- `git add`
- `git commit`
- `git log`
- `git status`

git init

- Create a new git repository

```
Ω ~/git_example/ git init
```

```
Initialized empty Git repository in /home/chad/git_example/.git/
```

git add

- Tell git that a newly created/changed file should be included in the next “commit” (snapshot)
- Sometimes called “staging” a change

```
Ω ~/git_example/ git add hello.py
```

git commit

- Save a snapshot of the current version of the code
- Include a human-readable message

```
Ω ~/git_example/ git commit -m "Created hello.py"  
[master (root-commit) 5c8c6ff] Created hello.py  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 hello.py
```

git commit

- Save a snapshot of the current version of the code
- Include a human-readable message

```
Ω ~/git_example/ git commit -m "Created hello.py"  
[master (root-commit) 5c8c6ff] Created hello.py  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 hello.py
```


git log

- View the commit history

```
Ω ~/git_example/ git log --oneline  
5c8c6ff (HEAD -> master) Created hello.py
```

git log

```
Ω ~/Documents/cs50/check50/ git log --oneline
f942cce (HEAD -> v3, origin/v3) update compiles method
baebff6 recurse submodules when fetching
5eb7e52 add update features
0588801 fix check50 case
6e70904 updated excepted branches and tags regex
61b8c21 Fix branch in travis badge
ac7be31 fix capitalization in index
df1cbf3 add check50.py to docs
611c1cb tweak flask exception message
bc08969 enable --local by default
be406d4 slightly improve docs
4b26d2b renamed process + updated docs
cf29235 improve docs + remove diff
```

git status

- View the current state of the repository
- Show modified, staged, and untracked files

```
Ω ~/git_example/ git status
```

```
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:   hello.py
```

```
Untracked files:
```

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

```
    goodbye.py
```

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

git status

- View the current state of the repository
- Show modified, staged, and untracked files

```
Ω ~/git_example/ git add hello.py
```

```
Ω ~/git_example/ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    modified:   hello.py
```

```
Untracked files:
```

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

```
    goodbye.py
```

Now, you try!

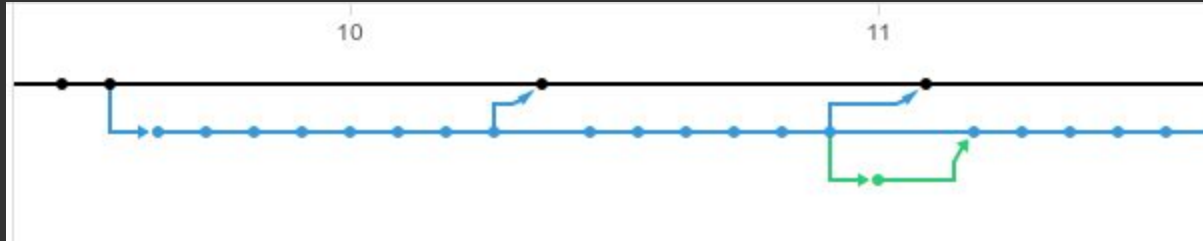
1. Open up a terminal in your Project 1 folder
2. Initialize a git repository
3. Add the files you want to commit
4. Create your first commit (don't forget to specify a message with `-m`)
5. View the log

If curious, run `git status` between each of the above steps.

Keeping multiple versions of your code

- `git branch`
- `git checkout`
- `git merge`

Git branches



<https://github.com/cs50/style50/network>

git branch

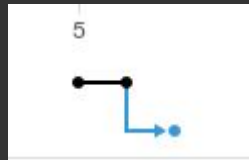
- List, create, and delete branches

```
Ω ~/git_example/ git branch
* master
```


git branch

- List, create, and delete branches

```
Ω ~/git_example/ git branch name
Ω ~/git_example/ git branch
* master
name
```



git checkout

- Switch to another branch

```
Ω ~/git_example/ git branch
```

```
* master
```

```
name
```

```
Ω ~/git_example/ git checkout name
```

```
Switched to branch 'name'
```

```
Ω ~/git_example/ git branch
```

```
master
```

```
* name
```

git merge

- Join two branches together, merging their commit histories

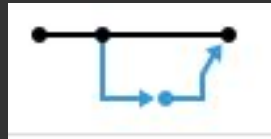
```
Ω ~/git_example/ git merge name
```

```
Updating bbce4eb..d94257e
```

```
Fast-forward
```

```
hello.py | 3 ++-
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```



Reverting changes

- `git reset`
- `git checkout (continued)`

git reset

- Returns code to a previous state
- Careful, cannot be undone!

```
Ω ~/git_example/ cat hello.py
name = input("Enter name: ")
print(f"hello, {name}")
Ω ~/git_example/ git log --oneline
d94257e (HEAD -> master, name) ask for name
bbce4eb added print statement
5c8c6ff Created hello.py
Ω ~/git_example/ git reset --hard bbce4eb
HEAD is now at bbce4eb added print statement
Ω ~/git_example/ cat hello.py
print("hello, world!")
Ω ~/git_example/ git log --oneline
bbce4eb (HEAD -> master) added print statement
5c8c6ff Created hello.py
```

git checkout (continued)

- View the state of the repository at a particular commit

```
Ω ~/git_example/ git checkout bbce4eb
```

```
Note: checking out 'bbce4eb'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-b` with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

```
HEAD is now at bbce4eb added print statement
```

git checkout (continued)

To review:

```
$ git checkout <BRANCH_NAME>
```

switches to a different branch

```
$ git checkout <COMMIT_HASH>
```

allows you a (read-only) view of an earlier commit

Now, you try!

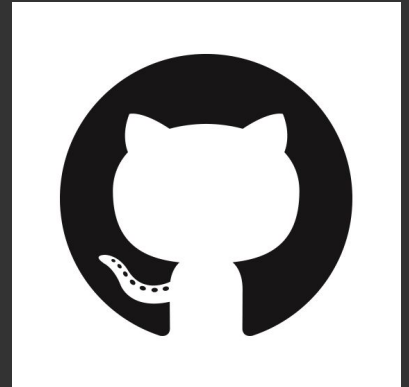
1. Create a new branch in your Project 1 repository
2. Make a small change (add a comment)
3. Merge this change back into master
4. Use `git reset` to revert this change

Collaborating with others

- `git remote`
- `git push`
- `git pull`

But first...

GitHub





Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



cs50 / style50

Unwatch 10

★ Star 4

Fork 4

Code

Issues 3

Pull requests 1

Insights

<https://manual.cs50.net/style50>

118 commits

5 branches

17 releases

5 contributors

MIT

Branch: develop

New pull request

Create new file

Upload files

Find file

Clone or download



crossroads1112 Fix json-mode error

Latest commit 7dc705a on May 30

style50

Fix json-mode error

a month ago

.gitignore

added .travis.yml

11 months ago

.travis.yml

added .travis.yml

11 months ago

LICENSE

Add LICENSE

11 months ago

README.md

Added --ignore to allow paths to be ignored when style50 runs recursi...

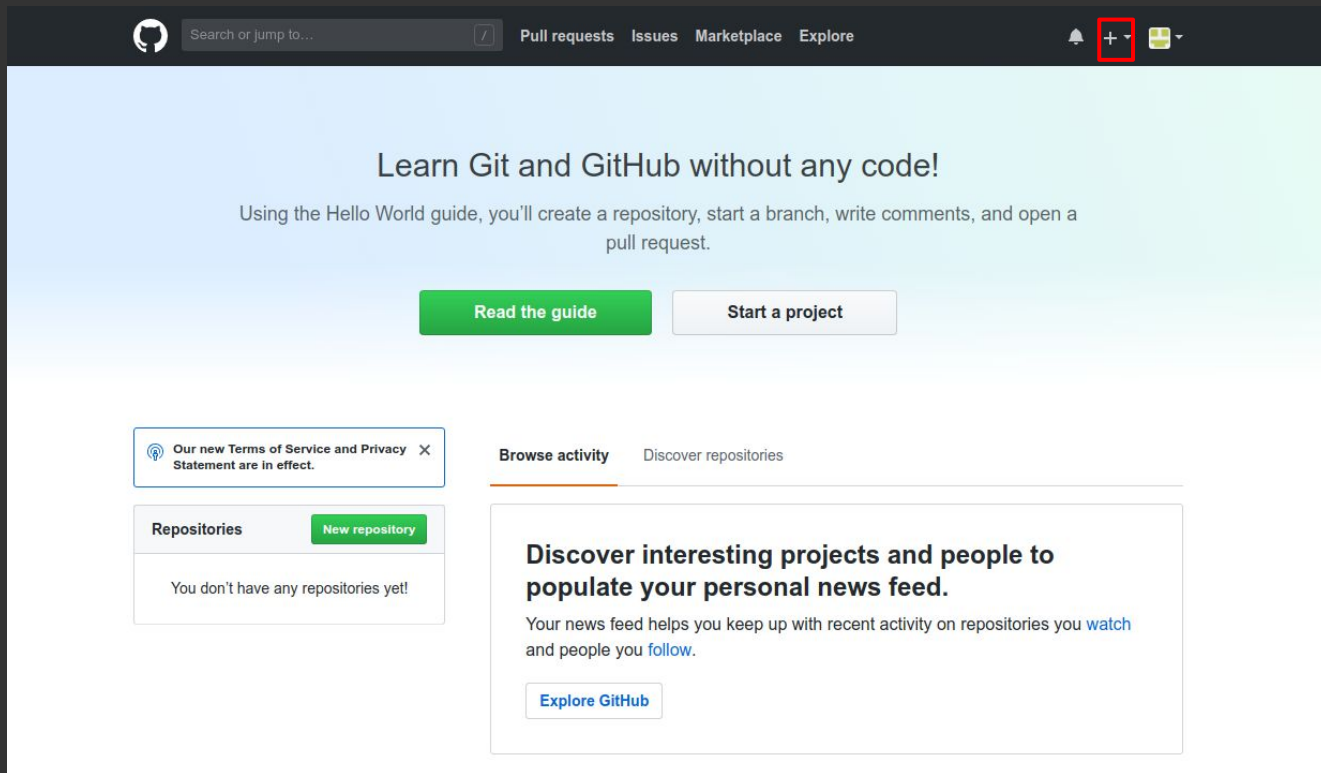
2 months ago

setup.py

version++

2 months ago

Create a new repository



The screenshot shows the GitHub homepage. At the top, there is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right side of the navigation bar, there is a bell icon, a red box highlighting a '+' icon, and a profile icon. Below the navigation bar, the main content area has a light blue background with the text "Learn Git and GitHub without any code!" and a subtext "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." There are two buttons: "Read the guide" (green) and "Start a project" (white). Below this, there is a notification banner for the new Terms of Service and Privacy Statement. On the left, there is a "Repositories" section with a "New repository" button and the text "You don't have any repositories yet!". On the right, there is a "Browse activity" section with the text "Discover interesting projects and people to populate your personal news feed." and a link to "Explore GitHub".

Search or jump to... Pull requests Issues Marketplace Explore

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide Start a project

Our new Terms of Service and Privacy Statement are in effect.

Repositories **New repository**

You don't have any repositories yet!

Browse activity Discover repositories

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

Explore GitHub

Create a new repository


Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 cmls3814 ▾

Repository name

project1 

Great repository names are short and memorable. Need inspiration? How about [miniature-fiesta](#).

Description (optional)

My website!



Public

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



Private


You choose who can see and commit to this repository.



Initialize this repository with a README

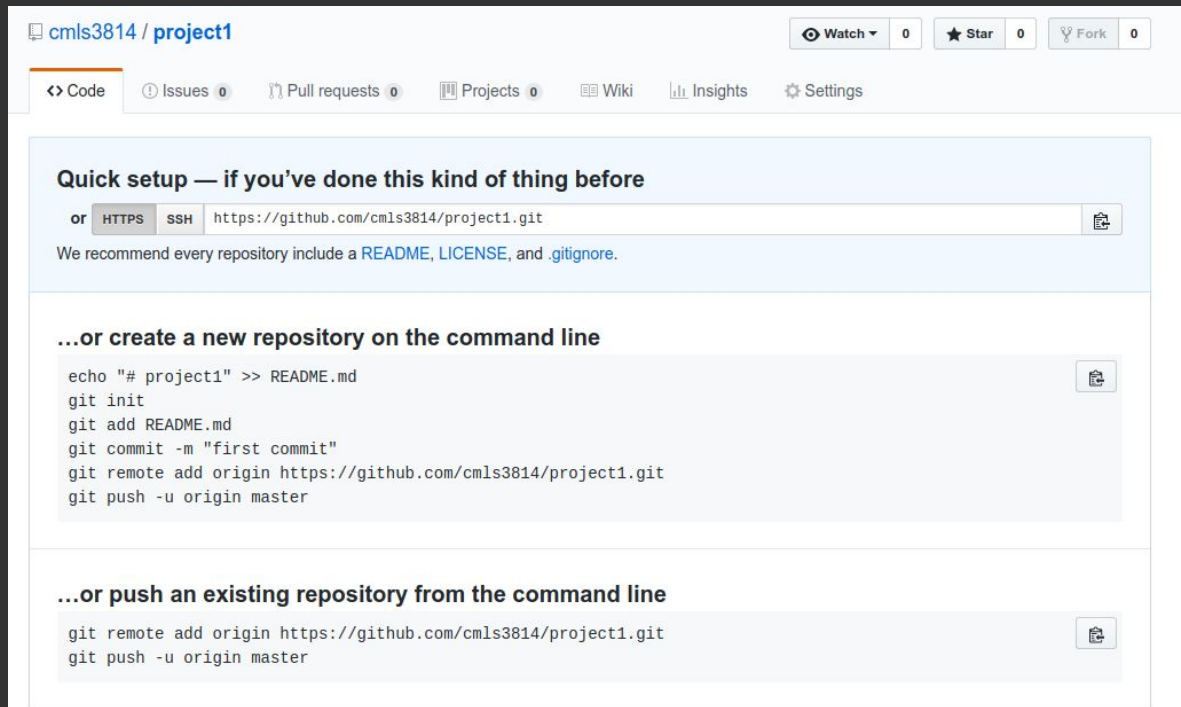
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

Create a new repository



The screenshot shows the GitHub interface for creating a new repository. At the top, the user's name 'cmls3814' and the repository name 'project1' are displayed. On the right, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area is titled 'Quick setup — if you've done this kind of thing before' and provides a URL for cloning the repository: `https://github.com/cmls3814/project1.git`. Below this, it recommends including a README, LICENSE, and .gitignore. The next section, '...or create a new repository on the command line', shows a series of terminal commands to create and push a new repository. The final section, '...or push an existing repository from the command line', shows the commands to push an existing local repository to the new GitHub repository.

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

or **HTTPS** **SSH** `https://github.com/cmls3814/project1.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# project1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/cmls3814/project1.git
git push -u origin master
```

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

```
git remote add origin https://github.com/cmls3814/project1.git
git push -u origin master
```

git remote

- Manage remote repositories

```
Ω ~/project1/ git remote add origin https://github.com/cmls3814/project1.git
Ω ~/project1/ git remote show origin
* remote origin
  Fetch URL: https://github.com/cmls3814/project1.git
  Push URL: https://github.com/cmls3814/project1.git
  HEAD branch: (unknown)
```


git push

- Update remote repository with any commits made locally

```
Ω ~/project1/ git push -u origin master
Username for 'https://github.com': cmls3814
Password for 'https://cmls3814@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/cmls3814/project1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

git pull

- Update local repository with any changes made remotely

```
Ω ~/project1/ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/cmls3814/project1
    1d4e126..e5d22b1  master      -> origin/master
Updating 1d4e126..e5d22b1
Fast-forward
 index.html | 3 +--
 1 file changed, 1 insertion(+), 2 deletions(-)
```

Now, you try!

- Create your Project 1 repository
- Add `https://github.com/<YOUR_USERNAME>/project1.git` as a remote repository
- Push your website to the remote repository

Congrats! You've created your first open-source project.

Oh no, merge conflicts!

Contributing to other repositories

- Filing issues on Github
- Forking a repository on Github
- `git clone`
- Creating pull requests

Issues

- Bug reports/feature requests

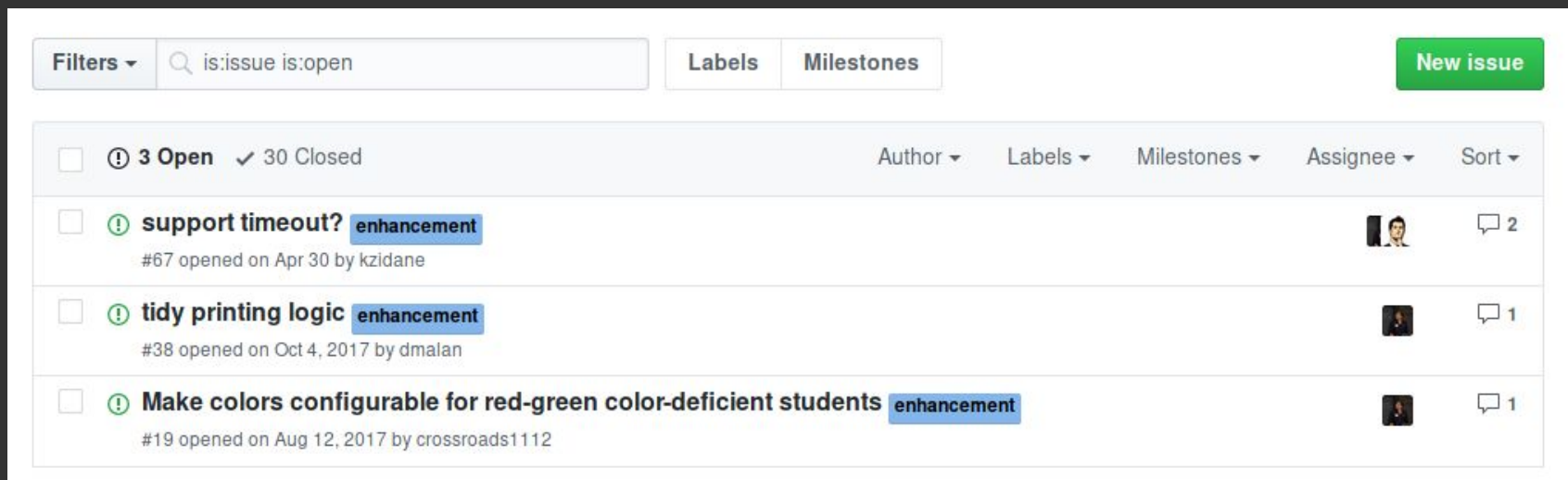
cs50 / style50

Watch 10 Star 4 Fork 5




Code Issues 3 Pull requests 1 Insights

<https://manual.cs50.net/style50>

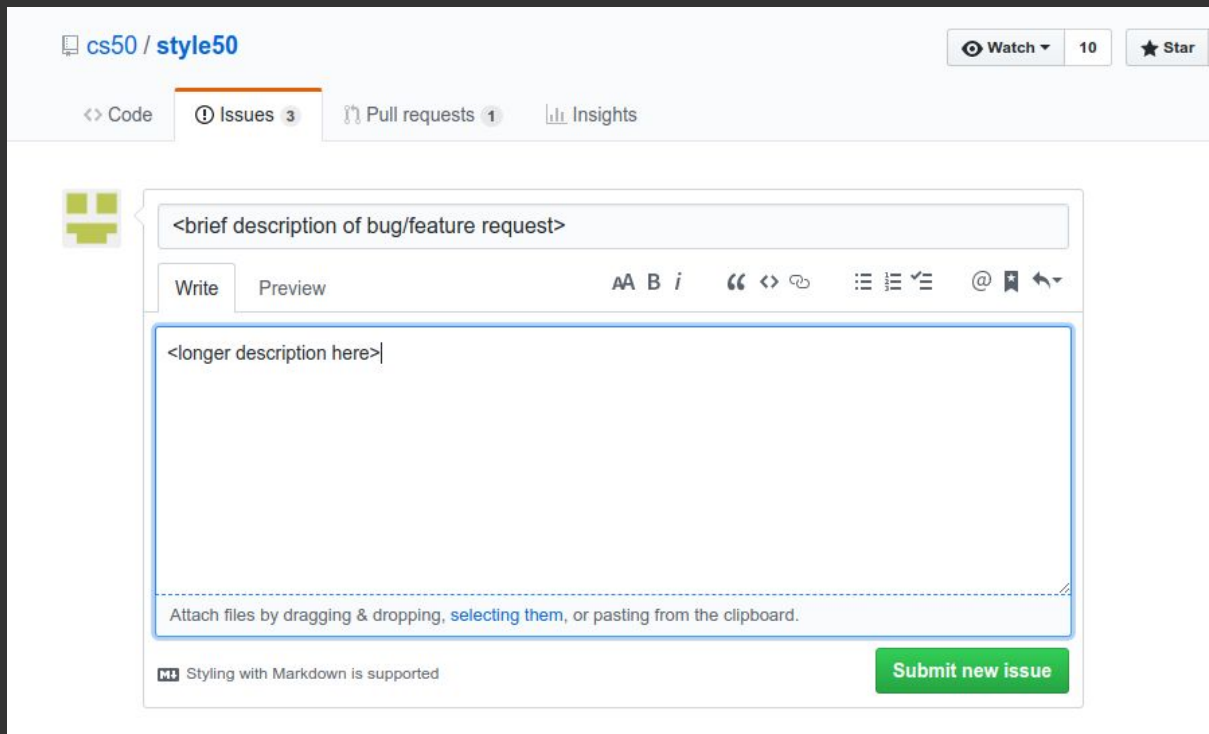
Filing issues on GitHub



The screenshot shows the GitHub Issues interface. At the top, there is a search bar with the text "is:issue is:open" and a "Filters" dropdown. To the right are "Labels" and "Milestones" buttons, and a green "New issue" button. Below this is a summary bar showing "3 Open" and "30 Closed" issues, along with sorting options for Author, Labels, Milestones, Assignee, and Sort. The main content area lists three open issues, each with a title, a label, the issue number, the opening date, the author, and the number of comments.

<input type="checkbox"/>	3 Open ✓ 30 Closed	Author	Labels	Milestones	Assignee	Sort
<input type="checkbox"/>	support timeout? enhancement #67 opened on Apr 30 by kzidane					2
<input type="checkbox"/>	tidy printing logic enhancement #38 opened on Oct 4, 2017 by dmalan					1
<input type="checkbox"/>	Make colors configurable for red-green color-deficient students enhancement #19 opened on Aug 12, 2017 by crossroads1112					1

Filing issues on GitHub

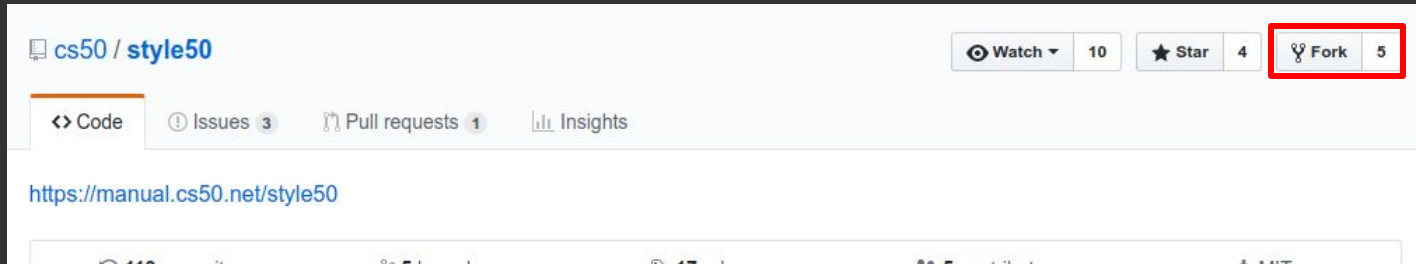


The screenshot shows the GitHub interface for filing an issue in the repository `cs50 / style50`. At the top right, there are buttons for "Watch" (with a dropdown arrow) and "Star" (with a star icon and the number "10"). Below this, navigation tabs include "Code", "Issues 3" (which is highlighted with an orange underline), "Pull requests 1", and "Insights".

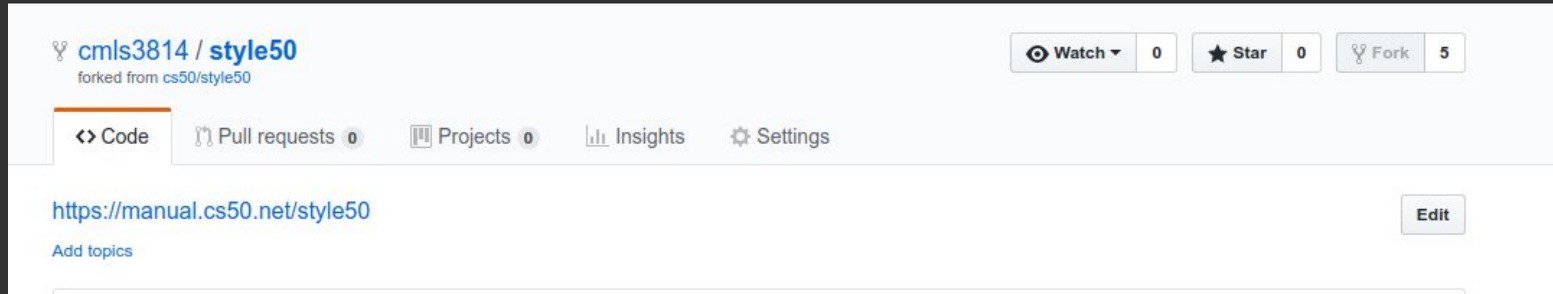
The main content area features a GitHub logo on the left and a form for creating a new issue. The form has a title input field containing the placeholder text "<brief description of bug/feature request>". Below the title field are two tabs: "Write" (selected) and "Preview". To the right of these tabs is a rich text editor toolbar with icons for bold (AA), italic (i), quote (left and right), code (code symbol), link (chain link), list (bulleted and numbered), and undo/redo (curved arrows).

The main text area of the form contains the placeholder text "<longer description here>". Below this text area, a dashed line separates it from a note that says "Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard." At the bottom left of the form, there is a small icon and the text "Styling with Markdown is supported". At the bottom right, there is a prominent green button labeled "Submit new issue".

Forking a repository on Github



Forking a repository on Github



The screenshot shows the GitHub interface for a repository named 'style50' owned by 'cmls3814'. The repository is a fork of 'cs50/style50'. The top navigation bar includes 'Watch' (0), 'Star' (0), and 'Fork' (5) buttons. Below this is a secondary navigation bar with 'Code', 'Pull requests' (0), 'Projects' (0), 'Insights', and 'Settings' tabs. The main content area displays the repository URL 'https://manual.cs50.net/style50' and an 'Edit' button. There is also a link to 'Add topics'.

cmls3814 / style50
forked from cs50/style50

Watch 0 Star 0 Fork 5

Code Pull requests 0 Projects 0 Insights Settings

<https://manual.cs50.net/style50> Edit

[Add topics](#)

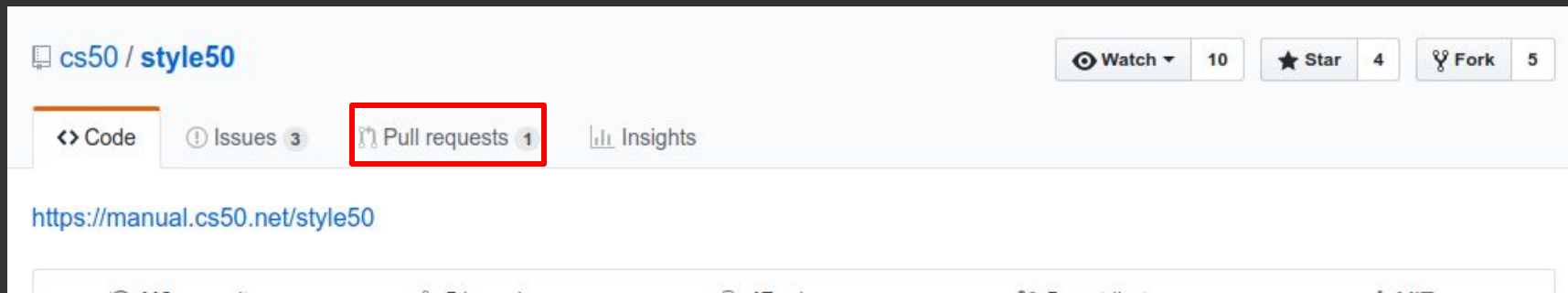
git clone

- Copy an existing remote repository locally

```
Ω ~/ git clone https://github.com/cmls3814/style50.git
Cloning into 'style50'...
remote: Counting objects: 495, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 495 (delta 4), reused 9 (delta 4), pack-reused 481
Receiving objects: 100% (495/495), 86.39 KiB | 2.33 MiB/s, done.
Resolving deltas: 100% (306/306), done.
```

Pull Requests

- Request that the original repository adopt the changes you made



Creating pull requests

The screenshot shows the GitHub interface for the repository `cs50 / style50`. At the top right, there are buttons for `Watch` (10), `Star` (4), and `Fork` (5). Below these are navigation tabs for `Code`, `Issues` (3), `Pull requests` (1), and `Insights`. The `Pull requests` tab is active and highlighted with an orange bar. Below the navigation is a search bar with the filter `is:pr is:open` and buttons for `Labels` and `Milestones`. A green `New pull request` button is on the right. The main content area shows a summary: `1 Open` and `37 Closed`. Below this is a table of pull requests with columns for `Author`, `Labels`, `Milestones`, `Reviews`, `Assignee`, and `Sort`. The first pull request is `Re-add javascript` with a green checkmark, opened 11 days ago by `crossroads1112`.

cs50 / style50

Watch 10 Star 4 Fork 5

Code Issues 3 Pull requests 1 Insights

Filters is:pr is:open Labels Milestones New pull request

1 Open 37 Closed Author Labels Milestones Reviews Assignee Sort

Re-add javascript ✓
#71 opened 11 days ago by crossroads1112

Creating pull requests

The screenshot shows the GitHub interface for a pull request comparison. At the top, the repository is identified as `cs50 / style50`. On the right, there are buttons for `Watch` (10), `Star` (4), and `Fork` (5). Below these, navigation tabs include `Code`, `Issues` (3), `Pull requests` (1), and `Insights`.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

The comparison interface shows:

- `base fork: cs50/style50`
- `base: develop`
- `head fork: cmls3814/style50`
- `compare: develop`

A green checkmark indicates: **✓ Able to merge.** These branches can be automatically merged.

A prominent green button labeled **Create pull request** is visible, with the text: *Discuss and review the changes in this comparison with others.*

Summary statistics at the bottom of the comparison area:

- 1 commit
- 1 file changed
- 0 commit comments
- 1 contributor

Below the comparison, a commit history section shows:

- Commits on Jul 05, 2018
- A commit by `crossroads1112` titled `update readme.md` by user `eeb5583`.

Why contribute to other projects?

cs50 / libcs50

Unwatch 31

★ Star 145

Fork 83

<> Code

🔔 Issues 3

🔗 Pull requests 1

📊 Insights

Fixed over/underflow and possible segfaults #1

Edit

Closed crossroads1112 wants to merge 9 commits into `cs50:master` from `unknown repository`

💬 Conversation 2

↔️ Commits 9

📄 Checks 0

📄 Files changed 4

+164 -69



crossroads1112 commented on Sep 10, 2015

Member



Reviewers



Now, you try!

1. Find a partner
2. Exchange links to your respective repositories
3. Fork and clone their repository into the IDE and try to improve it some way.
4. Push your changes to your fork, and create a pull request on their repository.

tinyurl.com/feedback14300

lunch!