

Git Workflow

The git workflow for contributing to open source repositories.

Git workflow

This document describes the git workflow that should be used when contributing to open source projects on GitHub. It assumes a very basic understanding of git (commits, branches, etc.) using the command line.

Note: This workflow is designed for open source (i.e., public) repositories. The workflow for private repositories may be slightly different, in particular, the repository's team may prefer for you to not fork the repo but rather push branches to it directly (although note that GitHub does allow you to fork private repositories that you have push access to and keeps the fork private).

For this document, we will suppose that you want to contribute a patch to [sympy/sympy](#).

NOTE: If you want to test out this guide without making an actual fix, please test against THIS REPO ([asmeurer/git-workflow](#)). Do not make a pull request against [sympy/sympy](#) unless you are making an actual change.

Cloning and forking the repository

Note: The steps in this section only need to be performed once per repository (e.g., if you contribute a second change to [sympy/sympy](#), you would not need to repeat them, but if you contribute to [sympy/sympy-live](#) you will need to repeat them for that repository).

1. **Clone the repository.** Click the green “Clone or download” button ①, and copy the url ② and type

```
git clone clone-url
```

at the terminal. Replace `clone-url` with the url that has been copied to your clipboard. For [sympy/sympy](#), it will be `git@github.com:sympy/sympy.git`. If you have not set up your ssh keys with GitHub, use the https url by first clicking the `https` button ③.

This repository Search Pull requests Issues Gist

sympy / sympy Unwatch 272 Unstar 3,470 Fork 1,731

Code Issues 2,249 Pull requests 409 Projects 0 Wiki Pulse Graphs Settings

A computer algebra system written in pure Python <http://sympy.org/> Edit

26,771 commits 1 branch 53 releases 514 contributors BSD-3-Clause

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

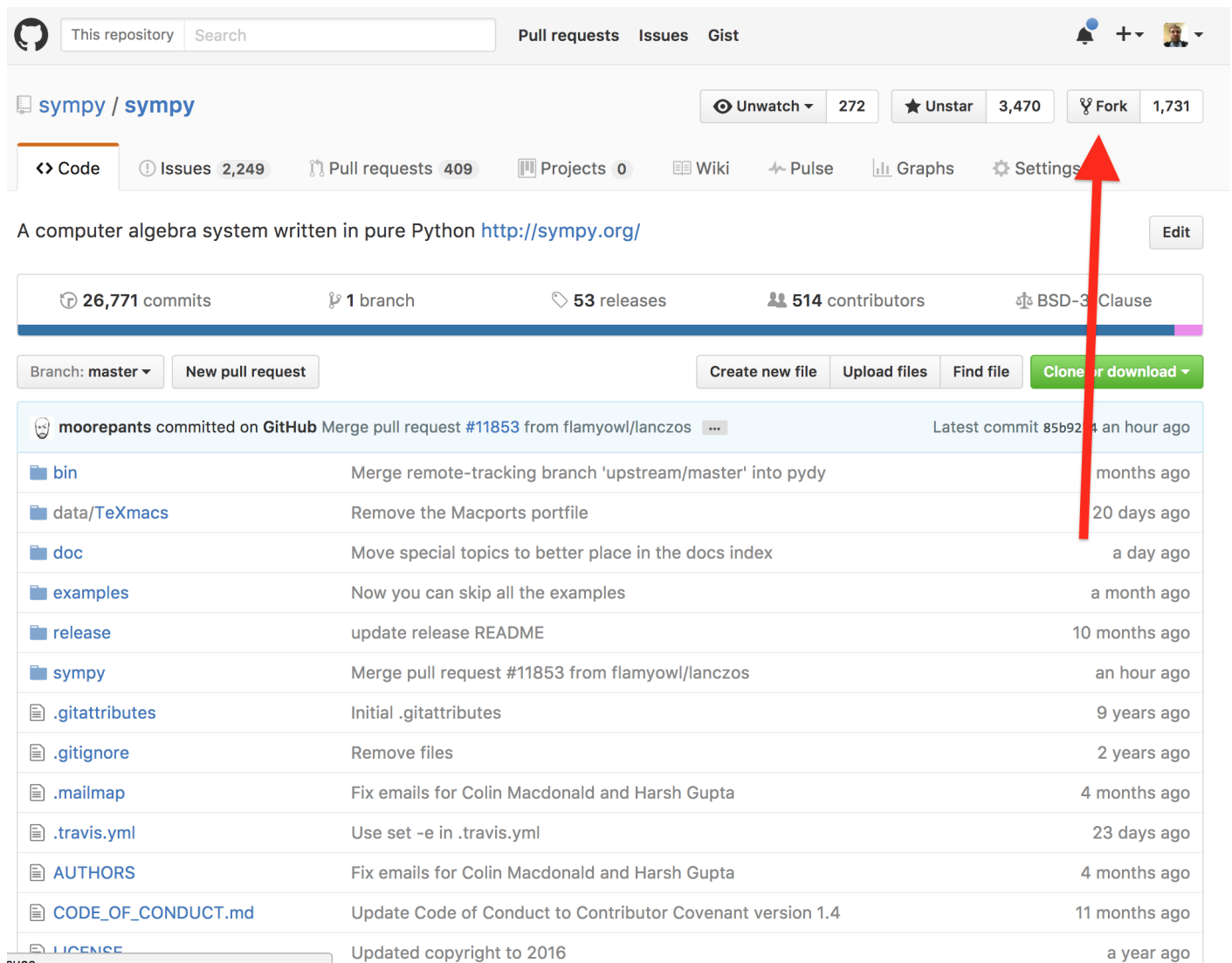
moorepants committed on GitHub Merge pull request #11853 from flamyowl/lanczos

bin	Merge remote-tracking branch 'upstream/master' into	
data/TeXmacs	Remove the Macports portfile	
doc	Move special topics to better place in the docs index	
examples	Now you can skip all the examples	
release	update release README	10 months ago
sympy	Merge pull request #11853 from flamyowl/lanczos	an hour ago
.gitattributes	Initial .gitattributes	9 years ago
.gitignore	Remove files	2 years ago
.mailmap	Fix emails for Colin Macdonald and Harsh Gupta	4 months ago
.travis.yml	Use set -e in .travis.yml	23 days ago
AUTHORS	Fix emails for Colin Macdonald and Harsh Gupta	4 months ago
CODE_OF_CONDUCT.md	Update Code of Conduct to Contributor Covenant version 1.4	11 months ago
LICENSE	Updated copyright to 2016	a year ago

Clone with SSH Use an SSH key and passphrase from account. git@github.com:sympy/sympy.git Open in Desktop Download ZIP Use HTTPS

Note: It is important that you clone from the repo you are contributing to (like sympy/sympy), not your fork of the repo.

2. **Fork the repo on GitHub to your personal account.** Click the **Fork** button on the sympy/sympy page.



The screenshot shows the GitHub repository page for `sympy / sympy`. At the top, there are navigation links for Pull requests, Issues, and Gist. Below the repository name, there are buttons for Unwatch (272), Unstar (3,470), and Fork (1,731). A red arrow points to the Fork button. Below the repository name, there are tabs for Code, Issues (2,249), Pull requests (409), Projects (0), Wiki, Pulse, Graphs, and Settings. The main content area shows a commit history table with columns for the commit message and the time since the commit. The table lists several commits, including a merge pull request #11853 from flamyowl/lanczos, and a commit to update the copyright to 2016.

Commit	Time
moorepants committed on GitHub Merge pull request #11853 from flamyowl/lanczos	Latest commit 85b9244 an hour ago
bin Merge remote-tracking branch 'upstream/master' into pydy	months ago
data/TeXmacs Remove the Macports portfile	20 days ago
doc Move special topics to better place in the docs index	a day ago
examples Now you can skip all the examples	a month ago
release update release README	10 months ago
sympy Merge pull request #11853 from flamyowl/lanczos	an hour ago
.gitattributes Initial .gitattributes	9 years ago
.gitignore Remove files	2 years ago
.mailmap Fix emails for Colin Macdonald and Harsh Gupta	4 months ago
.travis.yml Use set -e in .travis.yml	23 days ago
AUTHORS Fix emails for Colin Macdonald and Harsh Gupta	4 months ago
CODE_OF_CONDUCT.md Update Code of Conduct to Contributor Covenant version 1.4	11 months ago
LICENSE Updated copyright to 2016	a year ago

If you are presented with a list of organizations, click on your GitHub username.

3. **Add your fork as a remote.** This remote will be named after your github username. Go to the fork of your repository, in this case, `https://github.com/your-username/sympy` (replace `your-username` with your GitHub username), and copy the clone url as in step 1. `cd` to your clone from step 1 and run

```
git remote add your-github-username fork-url
```

(replace `your-github-username` with your GitHub username and `fork-url` with the url that was copied to your clipboard). You will be able to tell it is your fork url because it will have your GitHub username in it. For instance, if your username is `github_user`, you would run the command `git remote add github_user git@github.com:github_user/sympy.git`.

Remember, the above three steps only need to be performed once per repository. Once you have cloned and forked a repository once, there is no need to clone or fork it again.

Making changes

Before you make any changes, you should make a branch. Remember to **never commit to master**. The command `git status` will tell you what branch you are on. I recommend putting the git branch in your command prompt, so that you will always know what branch you are on. See [this guide](#) on how to do this.

It is important that you never commit to master because master will be the branch that you pull upstream changes from (e.g., changes from sympy/sympy).

1. **Update master.** Before you make any changes, first checkout master

```
git checkout master
```

and pull in the latest changes

```
git pull
```

This will make it so that your changes are against the very latest master, which will reduce the likelihood of merge conflicts due to your changes conflicting with changes made by someone else.

2. **Create a branch.** Once you have done this, create a new branch. You should make a branch name that is short, descriptive, and unique. Some examples of good branch names are `fix-install`, `docs-cleanup`, and `add-travis-ci`. Some examples of bad branch names are `feature`, `fix`, and `patch`. The branch name choice is not too important, so don't stress over it, but it is what people will use to reference your changes if they want to pull them down on their own computers to test them, so a good name will make it easier for others to understand what your branch does. In this example, the branch name is `fix-install`.

To create the branch, run

```
git checkout -b branch-name
```

(replace `branch-name` with the branch name you chose). This will create a new branch and check it out. You can verify this with `git status`.

3. **Make your changes and commit them.** Once you have created your branch, make your changes and commit them. Remember to keep your commits atomic, that is, each commit should represent a single unit of change. Also, remember to write helpful commit messages,

so that someone can understand what the commit does just from reading the message without having to read the diff.

For example, at the command line, this might look like

```
git add filename [filename ...]  
git commit
```

This will open an editor where you can write your commit message.

4. **Push up your changes.** Push your changes to your fork. Do this by running

```
git push your-github-username branch-name
```

(replace `your-github-username` with your GitHub username and `branch-name` with the name of the branch).

5. **Make a pull request.** If you then go to your fork on GitHub, you should see a button to create a pull request from your branch. It will look something like this:

This repository Search Pull requests Issues Gist

sympy / sympy Unwatch 272 Unstar 3,470 Fork 1,731

<> Code Issues 2,249 Pull requests 409 Projects 0 Wiki Pulse Graphs Settings

A computer algebra system written in pure Python <http://sympy.org/> Edit

26,771 commits 1 branch 53 releases 514 contributors BSD-3-Clause

Your recently pushed branches:

asmeurer:fix-install (less than a minute ago) Compare & pull request

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

Commit	Message	Time
moorepants committed on GitHub Merge pull request #11853 from flamyowl/lanczos		Latest commit 85b92b4 an hour ago
bin	Merge remote-tracking branch 'upstream/master' into pydy	3 months ago
data/TeXmacs	Remove the Macports portfile	20 days ago
doc	Move special topics to better place in the docs index	a day ago
examples	Now you can skip all the examples	a month ago
release	update release README	10 months ago
sympy	Merge pull request #11853 from flamyowl/lanczos	an hour ago
.gitattributes	Initial .gitattributes	9 years ago
.gitignore	Remove files	2 years ago
.mailmap	Fix emails for Colin Macdonald and Harsh Gupta	4 months ago
.travis.yml	Use set -e in .travis.yml	23 days ago

If you do not see this, go to the GitHub page for your fork ①, select the branch from the branch popup ② and click the pull request button ③.

The screenshot shows the GitHub repository page for `asmeurer / sympy`, which is a fork of `sympy/sympy`. The repository has 26,096 commits, 467 branches, 56 releases, 475 contributors, and is licensed under BSD-3-Clause. The repository description is "A computer algebra system written in pure Python. <http://sympy.org/>".

Below the repository information, the "Your recently pushed branches" section shows the `fix-install` branch, pushed less than a minute ago. A green button labeled "Compare & pull request" is next to it.

The "Switch branches/tags" dropdown menu is open, showing the `fix-install` branch selected. A red arrow points to this branch, labeled with a blue circle 2. Another red arrow points to the `master` branch in the "Branch:" dropdown, labeled with a blue circle 3.

The "Pull request" section shows a list of pull requests, including the one being created. The title field is highlighted with a red arrow and a blue circle 3.

Once doing this, you will be presented with a page. This page will show you the diff of the changes. Double check them to make sure you are making a pull request against the right branch.

Things to check here are that the base fork is the upstream repo ① (in this case, `sympy/sympy`) and the branch for the upstream repo is `master`, and that the head fork is your fork ② and the branch is the branch you wish to make the pull request from.

Enter a descriptive title in the title field ③. This is very important, as it is what will show up in the pull request listing and in email notifications to the people in the repo. Pull requests with undescriptive titles are more likely to be passed by. If the pull request fixes an issue, I recommend putting the issue number in the pull request description ④, not the title. People generally do not know issues by number, so a pull request that is just titled "fix for issue #1234" is more likely to be passed by, as it is unclear what it does from the title.

If there is more description or discussion about the pull request than what fits in the title field use the description field ④.

If the pull request fixes an issue, you can add “fixes #1234” (replace **1234** with the actual issue number) in the pull request description ④. This exact format, “fixes #1234” is important, as it will cause GitHub to automatically close the issue when the pull request is merged.

Once you are done, click the “create pull request” button ⑤.

base fork: sympy/sympy base: master ... head fork: asmeurer/sympy compare: fix-install

① Able to merge. These branches can be automatically merged.

②

Title of the pull request here ③

Write Preview AA B i “ < > ☰ ☷ ☹ ↶ @ 📎

Describe the changes in more depth here. ④

If this pull request fixes an issue indicate which issue by including "Fixes #1234" in this text.

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

☒ Allow edits from maintainers. Learn more ⑤

Create pull request

1 file changed 0 commit comments 1 contributor

6. Pushing additional changes. Once you have created the pull request, it will likely be reviewed and some additional fixes will be necessary. **Do not create a new pull request.** Rather, simply make more commits to your branch and push them up as in steps 3 and 4. They will be added to the pull request automatically. Note that although GitHub does notify people when you push new changes to a branch, many people have these notifications disabled as they can be quite noisy. So it is a good idea to make a comment on the pull request whenever you do so to notify people that it is ready to be reviewed again.

Once the pull request has been reviewed successfully, someone with push access to the main repository will merge it in. At this point you are done. You can checkout master and pull as described in step 1 and your changes should be there.

Important points

The important things to remember from this document are

1. You only need to clone and fork once per repository.
2. Always clone from the main repository and add your fork as a remote.
3. Never commit to master. Create a branch and commit to it.
4. Use `git status` often to check what branch you are on and see if you have any uncommitted changes.
5. Be descriptive in your branch names, commit messages, and pull request title and descriptions.
6. Once you have a pull request for a branch, you can push additional changes to the same branch and they will be added to the pull request automatically. You should never create a new pull request for the same branch.
7. Comment on the pull request when you want people to know that you have pushed new changes. Although GitHub does notify people of commit pushes, people are more likely notice your changes if you leave a comment.

[git-workflow](#) is maintained by [asmeurer](#).

This page was generated by [GitHub Pages](#).