Contributing to Astropy

=======================

Reporting Issues

----------------

When opening an issue to report a problem, please try to provide a minimal code

example that reproduces the issue along with details of the operating

system and the Python, NumPy, and `astropy` versions you are using.

Contributing

------------

So you are interested in contributing code to the Astropy Project? Excellent!

We love contributions! Astropy is open source, built on open source,

and we'd love to have you hang out in our community.

\*\*Imposter syndrome disclaimer\*\*: We want your help. No, really.

There may be a little voice inside your head that is telling you that you're not

ready to be an open source contributor; that your skills aren't nearly good

enough to contribute. What could you possibly offer a project like this one?

We assure you - the little voice in your head is wrong. If you can write code at

all, you can contribute code to open source. Contributing to open source

projects is a fantastic way to advance one's coding skills. Writing perfect code

isn't the measure of a good developer (that would disqualify all of us!); it's

trying to create something, making mistakes, and learning from those

mistakes. That's how we all improve, and we are happy to help others learn.

Being an open source contributor doesn't just mean writing code, either. You can

help out by writing documentation, tests, or even giving feedback about the

project (and yes - that includes giving feedback about the contribution

process). Some of these contributions may be the most valuable to the project as

a whole, because you're coming to the project with fresh eyes, so you can see

the errors and assumptions that seasoned contributors have glossed over.

Note: This disclaimer was originally written by

[Adrienne Lowe](https://github.com/adriennefriend) for a

[PyCon talk](https://www.youtube.com/watch?v=6Uj746j9Heo), and was adapted by

Astropy based on its use in the README file for the

[MetPy project](https://github.com/Unidata/MetPy).

### How to Contribute, Best Practices

All contributions to Astropy are done via [pull requests](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests) from GitHub users'

"forks" (i.e., copies) of the [astropy repository](https://github.com/astropy/astropy). If you

are new to this style of development, you will want to read over our

[development workflow](https://docs.astropy.org/en/latest/development/workflow/development\_workflow.html).

You may also/instead be interested in contributing to an

[astropy affiliated package](https://www.astropy.org/affiliated/).

Affiliated packages are astronomy-related software packages that are not a part

of the `astropy` core package, but build on it for more specialized applications

and follow the Astropy guidelines for reuse, interoperability, and interfacing.

Each affiliated package has its own developers/maintainers and its own specific

guidelines for contributions, so be sure to read their docs.

Once you open a pull request (which should be opened against the ``master``

branch, not against any of the other branches), please make sure to

include the following:

- \*\*Code\*\*: the code you are adding, which should follow

our [coding guidelines](https://docs.astropy.org/en/latest/development/codeguide.html) as much as possible.

- \*\*Tests\*\*: these are usually tests to ensure code that previously

failed now works (regression tests), or tests that cover as much as possible

of the new functionality to make sure it does not break in the future and

also returns consistent results on all platforms (since we run these tests on

many platforms/configurations). For more information about how to write

tests, see our [testing guidelines](https://docs.astropy.org/en/latest/development/testguide.html).

- \*\*Documentation\*\*: if you are adding new functionality, be sure to include a

description in the main documentation (in ``docs/``). Again, we have some

detailed [documentation guidelines](https://docs.astropy.org/en/latest/development/docguide.html) to help you out.

- \*\*Performance improvements\*\*: if you are making changes that impact `astropy`

performance, consider adding a performance benchmark in the

[astropy-benchmarks](https://github.com/astropy/astropy-benchmarks)

repository. You can find out more about how to do this

[in the README for that repository](https://github.com/astropy/astropy-benchmarks#contributing-benchmarks).

- \*\*Changelog entry\*\*: whether you are fixing a bug or adding new

functionality, you should add an entry to the [``CHANGES.rst``](CHANGES.rst) file that

includes the PR number. If you are opening a pull request you may not know

the PR number yet, but you can add it once the pull request is open. If you

are not sure where to put the changelog entry, wait until a maintainer

has reviewed your PR and assigned it to a milestone.

You do not need to include a changelog entry for fixes to bugs introduced in

the developer version and therefore are not present in the stable releases. In

general you do not need to include a changelog entry for minor documentation

or test updates. Only user-visible changes (new features/API changes, fixed

issues) need to be mentioned. If in doubt, ask the core maintainer reviewing

your changes.

Other Tips

----------

- Behind the scenes, we conduct a number of tests or checks with new pull requests.

This is a technique that is called continuous integration, and we use Travis CI

and CircleCI. To prevent the automated tests from running, you can add ``[ci skip]``

to your commit message. This is useful if your PR is a work in progress (WIP) and

you are not yet ready for the tests to run. For example:

$ git commit -m "WIP widget [ci skip]"

- If you already made the commit without including this string, you can edit

your existing commit message by running:

$ git commit --amend

- To skip only the tests running on Travis CI use ``[skip travis]``.

- If your commit makes substantial changes to the documentation but no code

changes, then you can use ``[skip travis]``, which will skip Travis CI

because documentation build is done on CircleCI. The exception to this rule

is when your changes to documentation include code snippets that need to

be tested using ``doctest``.

- When contributing trivial documentation fixes (i.e., fixes to typos, spelling,

grammar) that don't contain any special markup and are not associated with

code changes, please include the string ``[skip travis]`` in your commit

message.

$ git commit -m "Fixed typo [skip travis]"

Checklist for Contributed Code

------------------------------

A pull request for a new feature will be reviewed to see if it meets the

following requirements. For any pull request, an `astropy` maintainer can help

to make sure that the pull request meets the requirements for inclusion in the

package.

\*\*Scientific Quality\*\* (when applicable)

\* Is the submission relevant to astronomy?

\* Are references included to the origin source for the algorithm?

\* Does the code perform as expected?

\* Has the code been tested against previously existing implementations?

\*\*Code Quality\*\*

\* Are the [coding guidelines](https://docs.astropy.org/en/latest/development/codeguide.html) followed?

\* Is the code compatible with Python >=3.6?

\* Are there dependencies other than the `astropy` core, the Python Standard

Library, and NumPy 1.16.0 or later?

\* Is the package importable even if the C-extensions are not built?

\* Are additional dependencies handled appropriately?

\* Do functions that require additional dependencies raise an `ImportError`

if they are not present?

\*\*Testing\*\*

\* Are the [testing guidelines](https://docs.astropy.org/en/latest/development/testguide.html) followed?

\* Are the inputs to the functions sufficiently tested?

\* Are there tests for any exceptions raised?

\* Are there tests for the expected performance?

\* Are the sources for the tests documented?

\* Have tests that require an [optional dependency](https://docs.astropy.org/en/latest/development/testguide.html#tests-requiring-optional-dependencies)

been marked as such?

\* Does ``tox -e test`` run without failures?

\*\*Documentation\*\*

\* Are the [documentation guidelines](https://docs.astropy.org/en/latest/development/docguide.html) followed?

\* Is there a docstring in [numpydoc format](https://numpydoc.readthedocs.io/en/latest/format.html) in the function describing:

\* What the code does?

\* The format of the inputs of the function?

\* The format of the outputs of the function?

\* References to the original algorithms?

\* Any exceptions which are raised?

\* An example of running the code?

\* Is there any information needed to be added to the docs to describe the

function?

\* Does the documentation build without errors or warnings?

\*\*License\*\*

\* Is the `astropy` license included at the top of the file?

\* Are there any conflicts with this code and existing codes?

\*\*Astropy requirements\*\*

\* Do all the Travis CI and CircleCI tests pass?

\* If applicable, has an entry been added into the changelog?

\* Can you check out the pull request and repeat the examples and tests?