

Harmonize issue labels on GitHub across all repositories

Authors: François Michonneau, Tracy Teal

Date: 2018-02-20

Problem to solve

To improve the experience of contributing to and maintaining Carpentries lessons, and gain more meaningful contributions, it would be useful to have a consistent set of issue labels across all Carpentry repositories.

Historically, Software Carpentry has used a system of labels <http://swcarpentry.github.io/lesson-example/08-maintenance/index.html#issue-labels-in-repositories> across most of their repositories, but it hasn't been adopted by Data Carpentry. The merging of the two organizations, and plans for future expansion, provide a good opportunity to revisit this labelling system.

Considerations

Issue labels should:

- Be welcoming
- Be clear: help newcomers find what they can contribute to
- Be easy and useful for Maintainers to use and track
- Facilitate interaction with contributors by articulating why an issue won't be implemented
- State the type of change
- Identify when something is a discussion rather than a recommended change
- Identify things that need to be urgently addressed
- Be documented in the Carpentries handbook:
http://docs.carpentries.org/topic_folders/lesson_development/index.html (PR pending)
and a link to this page will be added to CONTRIBUTING.md
- Match labels used by the broader GitHub community where possible, example "help wanted"/"good first issue" are [advertised by GitHub](#)

Labels proposed

Status labels:

- **status:need-contributor**
 - **Hex code:** #DCECC7
 - **Short Description:** Looking for Contributors
 - **Long Description:** Issue reviewed by Maintainers, and ready to be addressed. Maintainers are looking for contributors to address this issue.
- **status:in-progress**
 - **Hex code:** #9BCC65
 - **Short Description:** Contributor working on issue
 - **Long Description:** A Contributor is actively working on addressing the issue, this label should be used once someone has been assigned the issue. Because, we can only assign people using GitHub's interface when they are part of the organization, the assignment is done by tagging them in a comment of the issue. The Maintainer should set an initial deadline for a PR to be submitted. We suggest 7 days, but it can be adapted to the discretion of the Maintainer depending on the complexity of the task.
- **status:pr-in-review**
 - **Hex code:** #679F38
 - **Short Description:** PR being reviewed by Maintainers
 - **Long Description:** A PR has been submitted to address this issue, and is under review by the Maintainers
- **status:completed**
 - **Hex code:** #33681E
 - **Short Description:** Issue successfully resolved
 - **Long Description:** One of several PRs have been merged, the issue is fully addressed.
 - **Note:** An issue or a pull request can be closed because it was not relevant to the lesson. We capture some of these reasons with the status labels below. Having a "completed" label does not fully duplicate the actions of "closing" an issue or "merging" a pull request. We need a way to capture issues that have been successfully addressed and resolved. If the issue has been opened, a PR has been associated with this issue, and the PR was merged, then we can use the GitHub API to infer that this issue was successfully addressed. However, there are edge cases that won't be captured by this process. These include: PRs that don't mention the issue, more than 1 PR needed to address the issue, or a change is required in another part of our infrastructure.
- **status:help-wanted**
 - **Hex code:** #FFF8C4
 - **Short Description:** Looking for Community Input
 - **Long Description:** Help from additional member of the Community is needed in order to address this issue.
- **status:need-more-info**

- **Hex code:** #EE6C00
- **Short Description:** More information needed
- **Long Description:** Issue doesn't include enough information to proceed
- **status:out-of-scope**
 - **Hex code:** #FFCC7F
 - **Short Description:** Issue or PR out of scope
 - **Long Description:** Issue that doesn't fall within the scope of the lesson
- **status:blocked**
 - **Hex code:** #E55100
 - **Short Description:** Progress on addressing issue blocked
 - **Long Description:** Issue can't be fixed because a technical problem is hindering progress. A Maintainer or someone else in the community should be notified to ensure that progress is being made.
- **status:refer-to-cac**
 - **Hex code:** #FFDFB2
 - **Short Description:** Curriculum Advisory Committee input needed
 - **Long Description:** Maintainers need advice from the Curriculum Advisory Committee to make a decision on how to proceed about addressing this issue
- **status:wait**
 - **Hex code:** #FFF2DF
 - **Short Description:** Progress dependent on other issue or conversation
 - **Long Description:** Issue dependent on another issue or ongoing conversation and cannot be addressed at this time.
- **status:duplicate**
 - **Hex code:** #BDBDBD
 - **Short Description:** Issue or PR already exists
 - **Long Description:** Issue raised previously, it should be linked to it by referring to it by its number in the comment before this label is used.

Type labels:

- **type:bug**
 - **Hex code:** #EB3F79
 - **Short Description:** Issue with code
 - **Long Description:** Issue about the code, including challenges, answers.
- **type:clarification**
 - **Hex code:** #00ACC0
 - **Short Description:** Issue or PR to make lesson clearer
 - **Long Description:** Part of a lesson which, while not incorrect (i.e., not a bug) is presented in a way that is potentially confusing or misleading. Existing content could benefit from rephrasing or rearranging.
- **type:discussion**

- **Hex code:** #B2E5FC
- **Short Description:** Issue to ask question
- **Long Description:** Issue used to ask a question about how the lesson is taught, ask for clarification. Such issues might indicate that the instructor guide or the documentation may need to be updated.
- **type:documentation**
 - **Hex code:** #4CB6AC
 - **Short Description:** Issue about auxiliary files
 - **Long Description:** Issue about something being outdated or inaccurate in the documentation of the process to building, or maintaining the lesson.
- **type:enhancement**
 - **Hex code:** #7FDEEA
 - **Short Description:** Propose enhancement to the lesson
 - **Long Description:** Proposal to add new content to the lesson (e.g., introducing additional function, library, command, flag), or adding more technical detail on a topic already covered in the lesson. Such issues may need to be considered by the infrastructure sub-committee, the curriculum advisory committee, or other relevant group.
- **type:feedback**
 - **Hex code:** #029AE5
 - **Short Description:** Issue to provide feedback on lesson
 - **Long Description:** General feedback on the lesson content
- **type:formatting**
 - **Hex code:** #AC1357
 - **Short Description:** Issue with lesson formatting
 - **Long Description:** Issue about something being wrong in the formatting of the lesson
- **type:instructor-guide**
 - **Hex code:** #00887A
 - **Short Description:** Issue with the instructor guide
 - **Long Description:** Issue related to the content of the instructor guide
- **type:teaching-example**
 - **Hex code:** #CED8DC
 - **Short Description:** PR showing how lesson was modified in a particular workshop
 - **Long Description:** PR that illustrates how someone modified the lesson in their workshop. Not intended to be merged, but as a way to document how other instructors have used the lesson.
- **type:template-and-tools**
 - **Hex code:** #7985CB
 - **Short Description:** Issue about template and tools

- **Long Description:** Issue or feature request about a technical aspect of the lesson (e.g., in the scripts used to render the lesson)
- **type:typo-text**
 - **Hex code:** #F8BAD0
 - **Short Description:** Typo in text
 - **Long Description:** Issue about a typo in the text/code of the lesson

Other labels:

- **Bug-bbq**
 - **Hex code:** #455964
 - **Short Description:** Good for Bug BBQ
 - **Long Description:** Good target for or related to Bug BBQ event
- **good first issue**
 - **Hex code:** #FFEB3A
 - **Short Description:** Good for new Contributors
 - **Long Description:** Good issue for a new Contributor
- **high-priority**
 - **Hex code:** #D22E2E
 - **Short Description:** Need to be addressed ASAP
 - **Long Description:** Issue that needs to be fixed as soon as possible because the lesson uses code that doesn't work anymore

Proposed workflow

1. Issue is opened on repository
2. Maintainer reviews issue and assigns a "type" label and a "status" label. Maintainers change the "status" label as Contributors address the issue. A regular progression of the status labels would be:
 - a. "Status:need-contributor" to indicate the issue is ready to be addressed. If not enough information is provided, Maintainers can use "Status:need-more-info" and ask the author of the issue or pull request for clarification.
 - b. Someone (including the author of the issue) manifests interest in addressing the issue, the Maintainer assigns the issue to this person by mentioning their name in the comment, sets a deadline for an initial PR, and changes the status label to "status:in progress"
 - c. If the deadline passes, and attempts to contact the person which was assigned the PR is unresponsive, the "status:available" label can be re-used.
 - d. If the person trying to implement the change faces challenges that can't be addressed easily, the label "status:blocked" can be used, and the Maintainer and/or the assigned person should try to find someone else in the community to help, or find a way to work around what is causing the issue.

Who will be affected by the proposed change

- Maintainers
- Lesson contributors

Who will lead and implement the proposed change

- François Michonneau

Once the sets of labels to use has been agreed on by the Maintainers and the infrastructure sub-committee, François will implement the changes as outlined below.

Timeline

- Identify repositories to try the issues labels on (See below)
- [Write script to push labels to the selected repositories -- By March 8th, 2018](#)
- Ask maintainers to put the tags on the issues themselves
- Ask Maintainers to deprecate old tags and replace with new ones when they go through the issues, get feedback about whether the new tags cover the scenarios
- Try new issue labels for a month until March 30th
- Get feedback from Maintainers of pilot repositories
- Deploy labels across all repositories by April 2nd
- Add labelling guidelines to Carpentry Handbook, and link there from Maintainer onboarding curriculum, and outline labels that affect contributors directly to CONTRIBUTING.md documentation
- Ask for feedback on labels after bug BBQ (April 16th)

Testing repositories:

- DC R-ecology <https://github.com/datacarpentry/R-ecology-lesson/issues/375>
- SWC python gapminder <https://github.com/swcarpentry/python-novice-gapminder/issues/286>
- SWC R gapminder <https://github.com/swcarpentry/r-novice-gapminder/issues/355>
- SWC git <https://github.com/swcarpentry/git-novice/issues/483>
- Instructor training: <https://github.com/carpentries/instructor-training/issues/>
- SWC Git Novice Spanish: <https://github.com/swcarpentry/git-novice-es/issues/>

Feedback received:

- Too many colors
- Rename status:help-wanted to status:expertise-needed to make distinction clearer between status:contributor-needed and status:help-wanted
- Too many “type:” labels
- status:completed not needed
- Rename out-of-scope to “long-term-idea”
- Why recreate/duplicate some of the standard GitHub labels?

References & Resources

- https://medium.com/@dave_lunny/sane-github-labels-c5d2e6004b63

Projects with interesting GitHub issue labels

- <https://github.com/akka/akka/labels>
- <https://github.com/matplotlib/matplotlib/labels>
- <https://github.com/pandas-dev/pandas/labels>
- <https://github.com/Data4Democracy/internal-displacement/labels>