

✓ 100 XP

Exercise - Create your first pull request

6 minutes

You've learned how to create a pull request (PR) when there's guidance either in a pull request template or in a CONTRIBUTING file. But what if a project doesn't offer that guidance and documentation on conventions?

Describe your changes

To write a good commit message, and subsequently your pull request, follow these practices:

- Your Git commit message subject line should complete the following sentence:
 - If applied, this commit will `<your subject line here>`.
- Include a succinct description of the change by using the imperative, present tense. For example, use *add* not *added* or *adds*.
- Limit your subject line to 50 characters.
- Start with a capital letter, and don't end with a period (.).
- You can use emojis in your subject line and `@mention` other GitHub users, but not everyone is a fan of such frivolity.

For the body of your message and pull request, continue to use present tense. Make sure to include the motivation for the change. Contrast your change with the previous behavior. Use the space at your disposal to explain the *what* and *why* versus the *how*.

Your commit message is only as much to the point as the content that you're about to submit. Commit or submit for review small, isolated sets of changes. This practice increases the likelihood of your changes getting merged into the project.

Add granularity

Before you submit your pull request, check the sidebar for ways to complete your PR. Select **Reviewers** or **Assignees** if you're familiar with the project's team structure. Add *labels* when there's guidance on using labels in, for instance, the CONTRIBUTING.md file. You can use labels as a visual clue for what you're trying to accomplish. A maintainer might also add a label or multiple labels.

Some of the labels we use in the repository for this Learn module are:

- **Bug** (red): Something isn't working
- **Documentation** (blue): Improvements or additions to documentation
- **Duplicate** (gray): This issue or pull request already exists
- **Enhancement** (teal): New feature or request

Optionally, you can *link issues* in the sidebar, where successfully merging a pull request might close the corresponding issue. You can also customize your subscription to *notifications* on the thread. Some PRs receive many comments, reviews, and CI/CD-related notifications. You can choose from:

- **Not subscribed**: Only receive notifications when you participated or were @mentioned
- **Subscribed**: Receive all notifications for this pull request
- **Custom**: Only be notified for the events you select

Exercise

Using the [First Contributions](#) project, practice forking, cloning, and submitting a pull request. The First Contributions project aims to "guide the way beginners make their first contribution." It has guides for both using the command line and several graphical user interfaces (GUIs). The project also has support for several languages. Make sure to check the `Translations` folder.

With the lessons from the previous unit and this one in mind, go back to a pull request you opened recently. Or, you can go to the pull requests tab of a project you're watching. Notice how a good subject line can make all the difference. Consider updating a pull request accordingly. Put roughly as much time into writing your PR as you did in making the change to the project. Your efforts will help the maintainers categorize and prioritize (triage) community contributions.

Bonus: Check Microsoft's Accessibility Guidelines and Requirements. In particular, see the information about [describing interactions with UI](#) to avoid ableist language in your contributions. Customers interact with products by using different input methods. For example, they can use the keyboard, a mouse, touch, voice, and more. You'll want to use generic verbs that work with any input method. For instance, use *select* instead of the input-specific *click* or *swipe*.

Next unit: Next steps

Continue >
