Tutorial content type

Tutorials are useful when someone has a basic understanding of the product and is interested in extending their understanding to solve a specific problem

> Articles in the "Contributing to GitHub Docs" section refer to the documentation itself and are a resource for GitHub staff and open source contributors.

Tutorials help people learn about products and solve real world problems by guiding them through the entire workflow to complete a task. Tutorials are more conversational in tone than other content. A tutorial feels like a developer-to-developer conversation while remaining accessible to readers with varied technical knowledge. Products with tutorials must already have a quickstart. For bite-sized workflows, use the quickstart model instead.

Tutorials are for people who want expert advice and a detailed discussion of best practices related to their problem. Tutorials also help people who've implemented similar solutions in the past with other products use GitHub. Tutorials can also help people validate whether the solution is appropriate for their needs.

We collectively refer to tutorials and quickstarts as "guides" across the site. On `/guides` landing pages, we include tutorials, quickstarts, and certain procedural articles in the list of guides for a doc set.

How to write a tutorial 🔗

For the tutorial template, see "[Templates](#)."

Contents of tutorials:

- Introduction

  - Clarifies audience.
  - Clearly states prerequisites and prior knowledge needed.
  - States what someone will accomplish or build.
  - Includes an example of a successful project.
  - Does not include the expected amount of time that it may take someone to complete the task - this depends on the experience level of the person completing the tutorial.

- Procedural sections

  - Based on the tutorial's audience, the steps can be less explicit and formal than those used in procedural content. You do not have to use existing reusables to

form these steps if the audience doesn't require that level of detail.

- Use: "From your profile, click **Settings**, and then click **Developer settings**."
- Avoid: In the upper-right corner of any page, click your profile photo, then click **Settings**. In the left sidebar, click **Developer settings**.

- Link out to other articles or resources rather than replicating them, to avoid interrupting the flow of information in the tutorial.
- Give visual cues. Use code blocks and screenshots heavily to help reassure people that they are performing the correct actions.
- Provide real examples.

  - For example, do not tell someone to "Enter a commit message" - instead, give them an appropriate example commit message that matches the previous steps.

- Troubleshooting

  - Acknowledge what may go wrong in the task and list a few common problems readers might run into with solutions.

- Conclusion

  - Review what was accomplished or built. Refer back to the project provided in the introduction as an example of a successful project.

- Next steps

  - Include 2-3 actionable next steps that someone can take after completing the tutorial. Link off to other related information like:

    - Projects on GitHub that illustrate the introduced concepts
    - Relevant information on docs.github.com
    - Relevant GitHub Skills
    - Relevant published talks, blog posts, or Community Forum series posts by Hubbers

Title guidelines for tutorials ⚭

- Follow the title guidelines for procedural articles.
- Do not use "tutorial" or "guide" in the title.

Examples of tutorials ⚭

Tutorials:

- [Adding labels to issues](#)
- [Installing an Apple certificate on macOS runners for Xcode development](#)

Language and framework guides:

- [Building and testing Node.js](#)
- [Building and testing Python](#)
- [Publishing Java packages with Maven](#)