

Creating tool switchers in articles

In this article

About tool switchers

Using tool tags

Adding new tools

You can use a tool switcher to show how to complete tasks using specific tools.

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

About tool switchers

In some articles, we write content tailored to different tools (the GitHub UI, GitHub CLI, GitHub Desktop, cURL, Codespaces, VS Code, GraphQL API, etc.) The tool switcher lets people select a tool to see only the content that is relevant to that tool since tools may have different conceptual or procedural information. People can use the tool switcher in two ways when reading the docs.

Exploring For tasks that can be completed with different tools, the tool switcher signals to people that there are multiple ways a task can be done. For example, using the GitHub CLI or GitHub Desktop, instead of the GitHub UI.

Getting to the point When someone knows how they want to do a task and doesn't need to see additional options, the tool switcher removes less relevant content, so they can find exactly what they need.

Using tool tags

We use tool tags to divide information for each tool. On rare occasions, we will add new tools.

Tool tags are a key value pair. The key is the tag you use to refer to the tool in the article and the value is how the tool will be identified on the tool picker at the top of the article. The existing tools are in [lib/all-tools.js](#) in the GitHub Docs repository.

When to use tool tags

We only use tool tags if an article must have tool-specific information to help people accomplish their tasks. If the conceptual information or procedural steps for a task are significantly different depending on what tool someone uses, and we want people to be able to accomplish the task with different tools, we use tool tags to present the relevant information in an article.

Do not use the tool switcher just to show examples in different languages. Only use the tool switcher if the tasks or concepts described in an article change based on what tool

someone uses.

How to use tool tags

Tool tags are Liquid tags that wrap content specific to a tool.

Put tools in alphabetical order. By default, the first tool tag will be selected for an article. You can define a different default tool for an article by specifying a `defaultTool:` property in the article's frontmatter. For more information, see the [content README](#).

You can also link to an article with a specific tool selected by adding `?tool=TOOLNAME` to the end of the link. For more information, see "[Style guide](#)."

Only include a maximum of eight different tools in an article. Including more tools causes the tool switcher tabs to overflow with an article's table of contents, which prevents people from using either the tool switcher or table of contents. It is unlikely that you will ever need to include eight separate tools in an article. In general, plan to use as few separate tools as possible in an article.

Adding new tools

GitHub Docs documents and maintains tool tags for GitHub products, GitHub-developed tools, and select third-party extensions.

New tools are only added when they are the only way to accurately document something for a specific user need. If a writer determines that adding a new tool is the only way to accurately document something, they need to propose the new tool in a content design plan. Whoever reviews the content design plan should consider any alternative ways to address the documentation needs without adding a new tool. If a new tool is the only way to create accurate documentation, the new tool should be added. If there is an alternative content solution that does not add a new tool, that option should be used.

To add a new tool, add an entry to the `allTools` object in the [lib/all-tools.js](#) file as a key-value pair. Add new tools in alphabetical order.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)