

When a new feature or integration is released that Gatsby developers can take advantage of, documentation should be added to improve the learning experience. Gatsby also needs docs for topic areas that exist in other technologies but are under-documented or under-tooled in the Gatsby ecosystem.

The knowledge of how to work with a technique, source plugin, or varied use case may be known internally to Gatsby team members, but it also may only exist on the web at large. This contributing doc is intended to provide a written process for producing docs without prior information, a critical function of the Gatsby Documentation team and open source community.

Identifying a topic

When identifying a topic, start by:

1. Looking at GitHub issues. Good labels to seek out are [help wanted](#) and [good first issue](#).
 - This part may be covered if a docs issue is assigned to you, or if you're signing up voluntarily to take on an issue.
2. Read through the existing Gatsby docs information and find gaps in topic coverage. Is there an area you feel is missing? [File an issue](#) to discuss it. If the team determines it warrants documentation, implement in a PR.
3. Observe common points of confusion or rough edges through user feedback and recommend solutions.

Note: It's required to open a GitHub issue before submitting a PR if one does not already exist.

Selecting the correct format

A GitHub issue for new learning material should indicate the format. Is it a Reference or Conceptual Guide? A Tutorial? A How-To Guide? Refer to the [Docs Structure](#) page for more information about the different documentation types.

Does docs coverage exist anywhere on `gatsbyjs.com`? If so, would an alternative format help provide information for Gatsby learners of different skill and experience levels? For example, if a tutorial exists but there is no coverage in Reference Guides, adding more content in a different format would benefit users.

Please follow the [Gatsby blog post guidelines](#) and do not suggest blog posts when what is needed is user documentation.

Gathering and validating supporting information

Writing an effective doc that meets the needs of Gatsby users requires gathering information from various sources and applying those concepts to your original writing. You must digest and validate the details you uncover and understand them enough to describe them in a way that users of multiple skill levels can learn.

Here are some tips for gathering information on a given topic within Gatsby:

1. As you prepare a contribution that adds documentation information, carefully read the accompanying GitHub issue for tips and relevant materials, and ask questions there.
2. Search the Gatsby GitHub repo for additional tips, examples, starters, plugins, READMEs, and other information that could help you provide a learning resource that guides users through a particular use case or concept.

3. You can also search the web for additional examples outside of the Gatsby GitHub org or docs site. Be sure to check Gatsby versions and only reference the most current examples unless for a specific purpose.
4. In the event there is no information available to write a greenfield doc after trying all the above steps, such as for a new integration, try asking Gatsby core team members to help to produce an outline and content tips. Please open a [documentation issue](#) for this then.

After you've collected supporting information, you must produce original writing to be accepted in the Gatsby docs. Copying other blog posts, materials, or Gatsby team member interviews word-for-word without attribution is not acceptable or allowed. Furthermore, direct quotes from interviews are also *almost never* effective for guides, recipes, or tutorials.

Rather, the best way to write greenfield docs is with new text that explains the concepts and ideas you uncovered. With research and demos supporting your writing, you can speak more legitimately to use cases and common troubleshooting. Phrasing information in alternative ways for different skill levels is a bit like avoiding usage of the same word or phrase in its definition: if a reader didn't understand it the first time, they may be more likely to understand it with an alternative explanation.

Producing learning materials

Create demo sites and examples to provide more authoritative material that supports the developer experience. When relevant, **functioning source code is a requirement to writing docs** that truly educate users. Source code examples can also be linked from a doc as an additional resource. Include testing as part of your source code to ensure it is robust and stands the test of time.

Follow the [docs structure](#) to ensure you're producing content in the right format for its purpose.

Use the [Markdown syntax doc](#) to understand your options for formatting text with Markdown, and follow [accessibility recommendations](#) for [heading levels](#) and image alt text.

Run the docs site locally to check formatting and functionality. There are instructions in the [contributing docs](#).

Refer to the [Gatsby Style Guide](#) to ensure your PR will be accepted.

Submitting a pull request

Submit a pull request that's tied to a GitHub issue by following the [How to Open a Pull Request guide](#).

Apply feedback from pull request reviews in order for them to be accepted. Further instructions can be found in the [How to Open a Pull Request guide](#).

Reference links

- [Docs Contributions](#)
- [Docs Structure](#)
- [How to File an Issue](#)
- [Gatsby Style Guide](#)
- [Markdown Syntax Doc](#)
- [Blog Post Guidelines](#)
- [Docs site setup instructions](#)
- [How to Open a Pull Request](#)