

Ways to contribute

So you want to contribute to Filecoin and the ecosystem? Here is a quick listing of things to which you can contribute and an overview on how you can get started.

Ways to contribute

Code

Filecoin and its sister-projects are big, with lots of code written in multiple languages. We always need help writing and maintaining code, but it can be daunting to just jump in. We use the label `Help Wanted` on features or bug fixes that people can help out with. They are an excellent place for you to start contributing code.

The biggest and most active repositories we have today are:

- [filecoin-project/venus](#)
- [filecoin-project/lotus](#)
- [filecoin-project/rust-fil-proofs](#)
-

If you want to start contributing to the core of Filecoin, those repositories are a great place start. But the `Help Wanted` label exists in several related projects:

- [IPFS](#)
- [libp2p](#)
- [IPLD](#)
- [Multiformats](#)
-

Documentation

Filecoin is a huge project and undertaking, and with lots of code comes the need for lots of good documentation! However, we need a lot more help to write the awesome docs the project needs. If writing technical documentation is your area, any and all help is welcome!

Before contributing to the Filecoin docs, please read these quick guides; they'll save you time and help keep the docs accurate and consistent!

1. [Style and formatting guide](#)
2. [Writing guide](#)
- 3.

If you have never contributed to an open-source project before, or just need a refresher, take a look at the [contribution tutorial](#).

Community

If interacting with people is your favorite thing to do in this world, join the [Filecoin chat and discussion forums](#) to say hello, meet others who share your goals, and connect with other members of the community. You should also consider joining [Filecoin Slack](#).

Build Applications

Filecoin is designed for you to integrate into your own applications and services.

Get started by looking at the list of projects currently built on Filecoin. Build anything you think is missing! If you're unsure about something, you can join the chat and discussion forums to get help or feedback on your specific problem/idea. You can also join a Filecoin Hackathon, apply for a Filecoin Developer Grant or apply to the Filecoin accelerator program to support the development of your project.

- [Filecoin Hackathons](#)
- [Filecoin Developer Grants](#)
- [Filecoin Accelerator Program](#)
-

Protocol Design

Filecoin is ultimately about building better protocols, and the community always welcome ideas and feedback on how to improve those protocols.

- [filecoin-project/specs](#)
-

Research

Finally, we see Protocol Labs as a research lab, where YOUR ideas can become technologies that have a real impact on the world. If you're interested in contributing to our research, please reach out to research@protocol.ai for more information. Include what your interests are so we can make sure you get to work on something fun and valuable.

Writing guide

This guide explains things to keep in mind when writing for Filecoin's documentation. While the [grammar, formatting, and style guide](#) lets you know the rules you should follow, this guide will help you to properly structure your writing and choose the correct tone for your audience.

Walkthroughs

The purpose of a walkthrough is to tell the user how to do something. They do not need to convince the reader of something or explain a concept. Walkthroughs are a list of steps the reader must follow to achieve a process or function.

The vast majority of documentation within the Filecoin documentation project falls under the Walkthrough category. Walkthroughs are generally quite short, have a neutral tone, and teach the reader how to achieve a particular process or function. They present the reader with concrete steps on where to go, what to type, and things they should click on. There is little to no conceptual information within walkthroughs.

Goals

Use the following goals when writing walkthroughs:

Goal Keyword Explanation Audience General Easy for anyone to read with minimal effort. Formality Neutral Slang is restricted, but standard casual expressions are allowed. Domain Technical Acronyms and tech-specific language is used and expected. Tone Neutral Writing contains little to no emotion. Intent Instruct Tell the reader how to do something. Function or process

The end goal of a walkthrough is for the reader to achieve a very particular function. Installing the Filecoin Desktop application is an example. Following this walkthrough isn't going to teach the reader much about working with the decentralized web or what Filecoin is. Still, by the end, they'll have the Filecoin Desktop application installed on their computer.

Short length

Since walkthroughs cover one particular function or process, they tend to be quite short. The estimated reading time of a walkthrough is somewhere between 2 and 10 minutes. Most of the time, the most critical content in a walkthrough is presented in a numbered list. Images and GIFs can help the reader understand what they should be doing.

If a walkthrough is converted into a video, that video should be no longer than 5 minutes.

Walkthrough structure

Walkthroughs are split into three major sections:

1. What we're about to do.
2. The steps we need to do.
3. Summary of what we just did, and potential next steps.
- 4.

Conceptual articles

Articles are written with the intent to inform and explain something. These articles don't contain any steps or actions that the reader has to perform right now.

These articles are vastly different in tone when compared to walkthroughs. Some topics and concepts can be challenging to understand, so creative writing and interesting diagrams are highly sought-after for these articles. Whatever writers can do to make a subject more understandable, the better.

Article goals

Use the following goals when writing conceptual articles:

Goal Keyword Explanation Audience Knowledgeable Requires a certain amount of focus to understand. Formality Neutral Slang is restricted, but standard casual expressions are allowed. Domain Any Usually technical, but depends on the article.

Tone Confident and friendly The reader must feel confident that the writer knows what they're talking about. Intent Describe Tell the reader why something does the thing that it does, or why it exists. Article structure

Articles are separated into five major sections:

1. Introduction to the thing we're about to explain.
2. What the thing is.
3. Why it's essential.
4. What other topics it relates to.
5. Summary review of what we just read.
- 6.

Tutorials

When writing a tutorial, you're teaching a reader how to achieve a complex end-goal. Tutorials are a mix of walkthroughs and conceptual articles. Most tutorials will span several pages, and contain multiple walkthroughs within them.

Take the hypothetical tutorial Get up and running with Filecoin , for example. This tutorial will likely have the following pages:

1. A brief introduction to what Filecoin is.
2. Choose and install a command line client.
3. Understanding storage deals.
4. Import and store a file.
- 5.

Pages 1 and 3 are conceptual articles, describing particular design patterns and ideas to the reader. All the other pages are walkthroughs instructing the user how to perform one specific action.

When designing a tutorial, keep in mind the walkthroughs and articles that already exist, and note down any additional content items that would need to be completed before creating the tutorial.

Grammar and formatting

Here are some language-specific rules that the Filecoin documentation follows. If you use a writing service like [Grammarly](#) , most of these rules are turned on by default.

American English

While Filecoin is a global project, the fact is that American English is the most commonly used style of English used today. With that in mind, when writing content for the Filecoin project, use American English spelling. The basic rules for converting other styles of English into American English are:

1. Swap thes
2. for az
3. in words like categorize
4. and pluralize
5. .
6. Remove theu
7. from words like color
8. and honor
9. .
10. Swap tre
11. for ter
12. in words like center
13. .
- 14.

The Oxford comma

In a list of three or more items, follow each item except the last with a comma, :

Use Don't use One, two, three, and four. One, two, three and four. Henry, Elizabeth, and George. Henry, Elizabeth and George.

References to Filecoin

As a proper noun, the name "Filecoin" (capitalized) should be used only to refer to the overarching project, to the protocol, or to the project's canonical network:

Filecoin [the project] has attracted contributors from around the globe! Filecoin [the protocol] rewards contributions of data

storage instead of computation! Filecoin [the network] is currently storing 50 PiB of data!

The name can also be used as an adjective:

The Filecoin ecosystem is thriving! I love contributing to Filecoin documentation!

When referring to the token used as Filecoin's currency, the name `FIL` , is preferred. It is alternatively denoted by the Unicode symbol for an integral with a double stroke `₿`:

- Unit prefix: 100 `FIL`
- .
- Symbol prefix: `₿` 100
- .
- .

The smallest and most common denomination of `FIL` is the `attoFIL` (10^{-18} `FIL`).

The collateral for this storage deal is 5 `FIL`. I generated `₿100` as a storage provider last month!

Examples of discouraged usage:

Filecoin rewards storage providers with Filecoin. There are many ways to participate in the filecoin community. My wallet has thirty filecoins.

Consistency in the usage of these terms helps keep these various concepts distinct.

References to Lotus

Lotus is the main implementation of Filecoin. As such, it is frequently referenced in the Filecoin documentation. When referring to the Lotus implementation, use a capital `L`. A lowercase `l` should only be used when referring to the Lotus executable commands such as `lotus daemon`. Lotus executable commands should always be within code blocks:

...

Copy 1. Start the Lotus daemon:

```
shell lotus daemon
```

1. After your Lotus daemon has been running for a few minutes, use `lotus` to check the number of other peers that it is connected to in the Filecoin network:

```
shell lotus net peers
```

...

Acronyms

If you have to use an acronym, spell the full phrase first and include the acronym in parentheses() the first time it is used in each document. Exception: This generally isn't necessary for commonly-encountered acronyms like `IPFS` , unless writing for a stand-alone article that may not be presented alongside project documentation.

Virtual Machine (VM), Decentralized Web (DWeb).

Formatting

How the Markdown syntax looks, and code formatting rules to follow.

Syntax

The Filecoin Docs project follows the `GitHub Flavoured Markdown` syntax for markdown. This way, all articles display properly within GitHub itself.

Rules

We use the rules set out in the [VSCode Markdownlint](#) extension. You can import these rules into any text editor like Vim or Sublime. All rules are listed [within the Markdownlint repository](#) .

We highly recommend installing [VSCode](#) with the [Markdownlint](#) extension to help with your writing. The extension shows warnings within your markdown whenever your copy doesn't conform to a rule.

Style

The following rules explain how we organize and structure our writing. The rules outlined here are in addition to the [rules](#) found within the [Markdownlinter extension](#).

Text

The following rules apply to editing and styling text.

Titles

1. All titles follow sentence structure. Only names
2. and places
3. are capitalized, along with the first letter of the title. All other letters are lower-case:
- 4.

...

Copy

This is a title

Only capitalize names and places

The capital city of France is Paris

...

1. Every article starts with a front-matter
2. title and description:
- 3.

...

Copy

title: Example article description: This is a brief description that shows up in link teasers in services like Twitter and Slack.

This is a subtitle

Example body text.

...

In the above example title: serves as a

or # tag. There is only ever one title of this level in each article.

1. **Titles do not contain punctuation. If you have a question within your title, rephrase it as a statement:**
- 2.

...

Copy

What is Filecoin?

Filecoin explained

...

Bold text

Double asterisks** are used to define boldface text. Use bold text when the reader must interact with something displayed as text: buttons, hyperlinks, images with text in them, window names, and icons.

...

Copy In the **Login** window, enter your email into the **Username** field and click **Sign in**.

...

Italics

Underscores_ are used to define italic text. Style the names of things in italics, except input fields or buttons:

...

Copy Here are some American things:

- The *Spirit of St Louis*.
- The *White House*.
- The *United States Declaration of Independence*.

Try entering them into the **American** field and clicking **Accept**.

...

Quotes or sections of quoted text are styled in italics and surrounded by double quotes" :

...

Copy In the wise words of Winnie the Pooh_ "People say nothing is impossible, but I do nothing every day."_

...

Code blocks

Tag code blocks with the syntax of the core they are presenting:

...

Copy javascript console.log(error);

...

Output from command-line actions can be displayed by adding another codeblock directly after the input codeblock. Here's an example telling the use to run go version and then the output of that command in a separate codeblock immediately after the first:

...

Copy shell go version

```
plaintext go version go1.19.7 darwin/arm64
```

...

Command-line examples can be truncated with three periods... to remove extraneous information:

...

Copy shell lotus-miner info

```
shell Miner:t0103 SectorSize:16.0MiB ... Sectors:map[Committing:0Proving:0Total:0]
```

...

Inline code tags

Surround directories, file names, and version numbers between inline code tags` .

...

Copy Version1.2.0of the program is stored in~/code/examples. Openexporter.exeto run the program.

...

List items

All list items follow sentence structure. Onlynames andplaces are capitalized, along with the first letter of the list item. All other letters are lowercase:

1. Never leave Nottingham without a sandwich.
2. Brian May played guitar for Queen.
3. Oranges.
- 4.

List items end with a period. , or a colon: if the list item has a sub-list:

1. Charles Dickens novels:
2.
 1. Oliver Twist.
3.
 1. Nicholas Nickelby.
4.
 1. David Copperfield.
5. 4.
6. J.R.R Tolkien non-fiction books:
7.
 1. The Hobbit.
8.
 1. Silmarillion.
9.
 1. Letters from Father Christmas.
10. 4.
- 11.

Unordered lists

Use the dash character- for un-numbered list items:

...

Copy - An apple. - Three oranges. - As many lemons as you can carry. - Half a lime.

...

Special characters

Whenever possible, spell out the name of the special character, followed by an example of the character itself within a code block.

...

Copy Use the dollar sign``to enter debug-mode.

...

Keyboard shortcuts

When instructing the reader to use a keyboard shortcut, surround individual keys in code tags:

...

Copy Pressctrl+ctocopythehighlightedtext.

...

The plus symbol+ stays outside of the code tags.

Images

The following rules and guidelines define how to use and store images.

Alt text

All images contain alt text so that screen-reading programs can describe the image to users with limited sight:

...

Copy

...

[Previous](#) [The Filecoin project](#) [Next](#) [Basics](#)

Last updated 4 months ago