

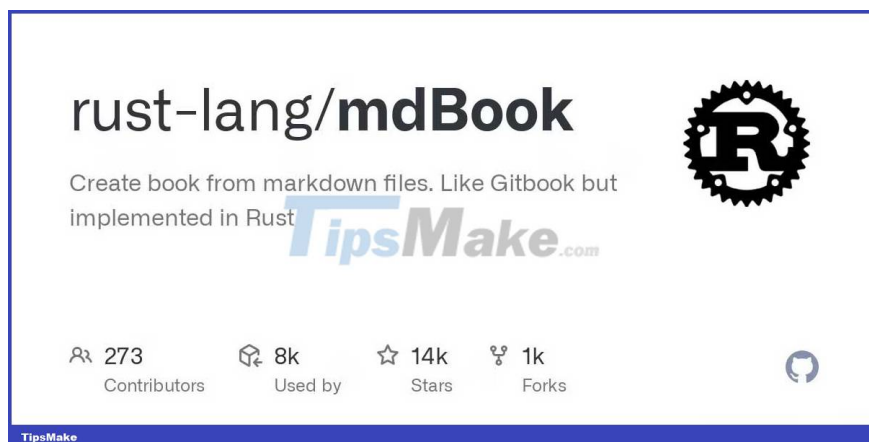
Documenting a Rust project with mdBook

Documentation plays an important role in the success of a project. It is a guide for programmers and users to better understand the project.



The **Rust** community recognizes the importance of comprehensive documentation for software development projects, so **Rust** has provided an official documentation tool, called **mdBook** .

What is mdBook?



mdBook is a free documentation tool for projects programming in Rust. It uses Markdown to create attractive and easy-to-navigate project documentation.

A major goal of documentation is to bridge the gap between code and human understanding. mdBook impresses by providing a structured format to make text easy to browse and find.

mdBook supports collaboration with a centralized knowledge sharing platform for stakeholders to contribute to documentation.

mdBook fosters teamwork, encourages the exchange of ideas and ensures a shared understanding of the project, improving the process of documenting as code. Collaborative methods increase productivity, reduce knowledge footprint, and enhance development workflows.

Get started with mdBook

mdBook is a command line tool that you can install via different sources.

mdBook is available on Cargo's registry package. If you have Rust and Cargo installed on your computer, you can use the **cargo install** command to install this command line tool.

```
cargo install mdbook
```

You can also install mdBook using Homebrew:

```
brew install mdbook
```

Once it's installed, you can use the **mdbook --version** command to verify the installation. This command prints the version of mdBook you have installed.

You can initialize an mdBook documentation project with the **init** command .

```
mdbook init my-docs
```

This example command creates a new folder named **my-docs** with the necessary file structure for the project.

mdBook uses a simple structure to organize documents:

```
. ??? book ??? book.toml ??? src ??? SUMMARY.md ??? chapter_1.md
```

Here is an overview of mdBook's document file structure:

1. **book/** : This directory contains the final output of the document.
2. **book.toml** : This is the configuration file for the documentation project. It allows you to define different settings and choices.
3. **src/** : This directory contains the source file for the document.
4. **SUMMARY.md** : This file acts as the table of contents for the document. It lists all the entries and sections.

You can use additional folders and configurations for your project's specific needs.

Create and organize items and sections

Open the **SUMMARY.md** file in a text editor and add these lines to the Markdown code:

```
# Table of Contents - [Introduction](chapters/introduction.md) - [Getting Started
```

You have added 3 sections to the document, including: Introduction, Getting Started, and Advanced Usage.

Create a src/chapters folder and create a Markdown file for each chapter inside the **chapters/** section .

You will document in a Markdown file for each chapter when writing a regular Markdown file.

Explain sample code for **chapters/advanced-usage.md** file .

```
# Advanced Usage This chapter will explore some advanced usage scenarios for our
? ?o?
n code Rust) ``rust use rayon::prelude::*; fn main() { let numbers = vec![1, 2,
?i ?ây, b?n ?ã nh?p crate rayon và dùng par_iter ?? l?p các s?
vector song song. B?n ?ã dùng ph??ng th?c sum ?? tính t?ng s? các ph?n t?
song song.
```

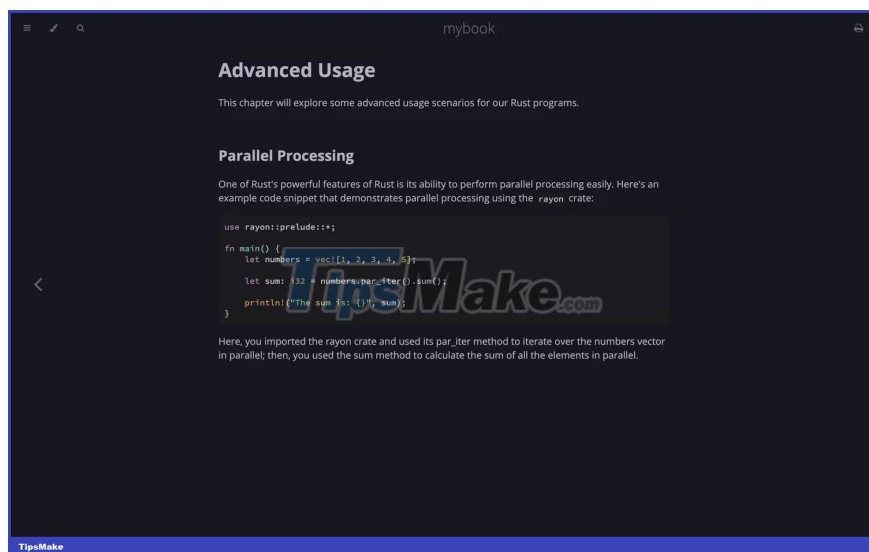
The Parallel Processing section begins with the Markdown **#** syntax that defines the section name.

Remember to follow the conventional Markdown syntax for formatting content. mdBook supports most of the Markdown functionality, including lists, paragraphs, links, and more.

After you've written a document, you can use the **mdBook** commands to work on it. For example, you can use the **mdbook server** command to process documents.

```
mdbook serve
```

When you run this command, mdBook will process the project's document on localhost port 3000, so you can view it in your browser at <http://localhost:3000/>.



mdBook can improve the documentation workflow for Rust projects. Try it and see for yourself!

You finished reading the article "**Documenting a Rust project with mdBook**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
