Introduction

This book is the primary reference for the Rust programming language. It provides three kinds of material:

- Chapters that informally describe each language construct and their use.
- Chapters that informally describe the memory model, concurrency model, runtime services, linkage model, and debugging facilities.
- Appendix chapters providing rationale and references to languages that influenced the design.

Warning: This book is incomplete. Documenting everything takes a while. See the GitHub issues for what is not documented in this book.

Rust releases

Rust has a new language release every six weeks. The first stable release of the language was Rust 1.0.0, followed by Rust 1.1.0 and so on. Tools (rustc ,cargo , etc.) and documentation (Standard library , this book, etc.) are released with the language release.

The latest release of this book, matching the latest Rust version, can always be found at https://doc.rust-lang.org/reference/ . Prior versions can be found by adding the Rust version before the "reference" directory. For example, the Reference for Rust 1.49.0 is located at https://doc.rust-lang.org/1.49.0/reference/ .

WhatThe Reference is not

This book does not serve as an introduction to the language. Background familiarity with the language is assumed. A separate book is available to help acquire such background familiarity.

This book also does not serve as a reference to the standard library included in the language distribution. Those libraries are documented separately by extracting documentation attributes from their source code. Many of the features that one might expect to be language features are library features in Rust, so what you're looking for may be there, not here.

Similarly, this book does not usually document the specifics of rustc as a tool or of Cargo. rustc has its own book . Cargo has a book that contains a reference . There are a few pages such as linkage that still describe how rustc works.

This book also only serves as a reference to what is available in stable Rust. For unstable features being worked on, see the Unstable Book .

Rust compilers, including rustc , will perform optimizations. The reference does not specify what optimizations are allowed or disallowed. Instead, think of the compiled program as a black box. You can only probe by running it, feeding it input and observing its output. Everything that happens that way must conform to what the reference says.

Finally, this book is not normative. It may include details that are specific to rustc itself, and should not be taken as a specification for the Rust language. We intend to produce such a book someday, and until then, the reference is the closest thing we have to one.

How to use this book

This book does not assume you are reading this book sequentially. Each chapter generally can be read standalone, but will cross-link to other chapters for facets of the language they refer to, but do not discuss.

There are two main ways to read this document.

The first is to answer a specific question. If you know which chapter answers that question, you can jump to that chapter in the table of contents. Otherwise, you can presss or click the magnifying glass on the top bar to search for keywords related to your question. For example, say you wanted to know when a temporary value created in a let statement is dropped. If you didn't already know that the lifetime of temporaries is defined in the expressions chapter , you could search "temporary let" and the first search result will take you to that section.

The second is to generally improve your knowledge of a facet of the language. In that case, just browse the table of contents until you see something you want to know more about, and just start reading. If a link looks interesting, click it, and read about that section.

That said, there is no wrong way to read this book. Read it however you feel helps you best.

Conventions

Like all technical books, this book has certain conventions in how it displays information. These conventions are documented here.

- Statements that define a term contain that term initalics
- .
- Whenever that term is used outside of that chapter, it is usually a link to the section that has this definition.
- Anexample term
- is an example of a term being defined.
- Differences in the language by which edition the crate is compiled under are in a blockquote that start with the words "Edition Differences:" inbold
- .
- Edition Differences
- : In the 2015 edition, this syntax is valid that is disallowed as of the 2018 edition.
- Notes that contain useful information about the state of the book or point out useful, but mostly out of scope, information are in blockquotes that start with the word "Note:" inbold
- .
- Note
- : This is an example note.
- Warnings that show unsound behavior in the language or possibly confusing interactions of language features are in a special warning box.
- Warning: This is an example warning.
- Code snippets inline in the text are inside
- tags.
- Longer code examples are in a syntax highlighted box that has controls for copying, executing, and showing hidden lines in the top right corner.
- // This is a hidden line.
- fn
- main
- () {println!
- ("This is a code example"
-);
- }
- All examples are written for the latest edition unless otherwise stated.
- The grammar and lexical structure is in blockquotes with either "Lexer" or "Syntax" inbold superscript
- as the first line.
- Syntax
- ExampleGrammar
- :
- ~
- Expression
- |box
- Expression
- SeeNotation
- for more detail.

Contributing

We welcome contributions of all kinds.

You can contribute to this book by opening an issue or sending a pull request to the Rust Reference repository . If this book does not answer your question, and you think its answer is in scope of it, please do not hesitate tofile an issue or ask about it in thet-lang/doc stream on Zulip . Knowing what people use this book for the most helps direct our attention to making those sections the best that they can be. We also want the reference to be as normative as possible, so if you see anything that is wrong or is non-normative but not specifically called out, please alsofile an issue .

Mobile navigation buttons