

```
..... {#mdbook-help-container}

.... {#mdbook-help-popup}

## Keyboard shortcuts {#keyboard-shortcuts .mdbook-help-title}
```

```
<div>
```

Press [←]{.kbd} or [→]{.kbd} to navigate between chapters

Press [S]{.kbd} or [/]{.kbd} to search in the book

Press [?]{.kbd} to show this help

Press [Esc]{.kbd} to hide this help

```
</div>
```

```
.....
```

```
.....
```

```
..... {#body-container}
```

1. [The Rust Programming

Language](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/title-page.html){.active  
tabindex="0"}

2.

[Foreword](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/foreword.html){tabindex="0"}

3.

[Introduction](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch00-00-introduction.html){tabindex="0"}

4. **1. Getting**

Started](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch01-00-getting-started.html){tabindex="0"}

5. 1. **1.1.**

Installation](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch01-01-installation.html){tabindex="0"}

2. **1.2.** Hello,

World!](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch01-02-hello-world.html){tabindex="0"}

3. **1.3.** Hello,

Cargo!](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch01-03-hello-cargo.html){tabindex="0"}

6. **2.** Programming a Guessing

Game](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch02-00-guessing-game-tutorial.html){tabindex="0"}

7. **3.** Common Programming

Concepts](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-00-common-programming-concepts.html){tabindex="0"}

## 8. 1. **3.1. Variables and**

Mutability](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-01-variables-and-mutability.html){tabindex="0"}

## 2. **3.2. Data**

Types](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-02-data-types.html){tabindex="0"}

## 3. **3.3.**

Functions](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-03-how-functions-work.html){tabindex="0"}

## 4. **3.4.**

Comments](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-04-comments.html){tabindex="0"}

## 5. **3.5. Control**

Flow](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch03-05-control-flow.html){tabindex="0"}

## 9. **4. Understanding**

Ownership](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch04-00-understanding-ownership.html){tabindex="0"}

## 10. 1. **4.1. What is**

Ownership?](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/bo

ok/ch04-01-what-is-ownership.html){tabindex="0"}

## 2. **4.2. References and**

Borrowing](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch04-02-references-and-borrowing.html){tabindex="0"}

## 3. **4.3. The Slice**

Type](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch04-03-slices.html){tabindex="0"}

## 11. **5. Using Structs to Structure Related**

Data](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch05-00-structs.html){tabindex="0"}

### 12. 1. **5.1. Defining and Instantiating**

Structs](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch05-01-defining-structs.html){tabindex="0"}

### 2. **5.2. An Example Program Using**

Structs](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch05-02-example-structs.html){tabindex="0"}

### 3. **5.3. Method**

Syntax](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch05-03-method-syntax.html){tabindex="0"}

## 13. **6. Enums and Pattern**

Matching](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch06-00-enums.html){tabindex="0"}

14. 1. **6.1. Defining an**

Enum](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch06-01-defining-an-enum.html){tabindex="0"}

2. **6.2. The match Control Flow**

Construct](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch06-02-match.html){tabindex="0"}

3. **6.3. Concise Control Flow with if let and let**

else](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch06-03-if-let.html){tabindex="0"}

15. **7. Managing Growing Projects with Packages, Crates, and**

Modules](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-00-managing-growing-projects-with-packages-crates-and-modules.html){tabindex="0"}

16. 1. **7.1. Packages and**

Crates](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-01-packages-and-crates.html){tabindex="0"}

2. **7.2. Defining Modules to Control Scope and**

Privacy](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-02-defining-modules-to-control-scope-and-privacy.html){tabindex="0"}

### 3. **7.3.** Paths for Referring to an Item in the Module

Tree](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-03-paths-for-referring-to-an-item-in-the-module-tree.html){tabindex="0"}

### 4. **7.4.** Bringing Paths Into Scope with the use

Keyword](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-04-bringing-paths-into-scope-with-the-use-keyword.html){tabindex="0"}

### 5. **7.5.** Separating Modules into Different

Files](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch07-05-separating-modules-into-different-files.html){tabindex="0"}

## 17. **8.** Common

Collections](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch08-00-common-collections.html){tabindex="0"}

### 18. 1. **8.1.** Storing Lists of Values with

Vectors](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch08-01-vectors.html){tabindex="0"}

### 2. **8.2.** Storing UTF-8 Encoded Text with

Strings](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch08-02-strings.html){tabindex="0"}

### 3. **8.3.** Storing Keys with Associated Values in Hash

Maps](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch

08-03-hash-maps.html){tabindex="0"}

## 19. **Error**

Handling](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch09-00-error-handling.html){tabindex="0"}

### 20. 1. **Unrecoverable Errors with**

panic!](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch09-01-unrecoverable-errors-with-panic.html){tabindex="0"}

#### 2. **Recoverable Errors with**

Result](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch09-02-recoverable-errors-with-result.html){tabindex="0"}

#### 3. **To panic! or Not to**

panic!](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch09-03-to-panic-or-not-to-panic.html){tabindex="0"}

## 21. **Generic Types, Traits, and**

Lifetimes](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch10-00-generics.html){tabindex="0"}

### 22. 1. **Generic Data**

Types](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch10-01-syntax.html){tabindex="0"}

#### 2. **Traits: Defining Shared**

Behavior](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch10-02-traits.html){tabindex="0"}

### 3. [\*\*10.3.\*\*] Validating References with

Lifetimes](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch10-03-lifetime-syntax.html){tabindex="0"}

## 23. [\*\*11.\*\*] Writing Automated

Tests](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch11-00-testing.html){tabindex="0"}

### 24. 1. [\*\*11.1.\*\*] How to Write

Tests](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch11-01-writing-tests.html){tabindex="0"}

### 2. [\*\*11.2.\*\*] Controlling How Tests Are

Run](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch11-02-running-tests.html){tabindex="0"}

### 3. [\*\*11.3.\*\*] Test

Organization](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch11-03-test-organization.html){tabindex="0"}

## 25. [\*\*12.\*\*] An I/O Project: Building a Command Line

Program](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-00-an-io-project.html){tabindex="0"}

### 26. 1. [\*\*12.1.\*\*] Accepting Command Line



Arguments](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-01-accepting-command-line-arguments.html){tabindex="0"}

## 2. **12.2. Reading a**

File](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-02-reading-a-file.html){tabindex="0"}

## 3. **12.3. Refactoring to Improve Modularity and Error**

Handling](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-03-improving-error-handling-and-modularity.html){tabindex="0"}

## 4. **12.4. Developing the Library's Functionality with Test**

Driven

Development](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-04-testing-the-librarys-functionality.html){tabindex="0"}

## 5. **12.5. Working with Environment**

Variables](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-05-working-with-environment-variables.html){tabindex="0"}

## 6. **12.6. Writing Error Messages to Standard Error Instead of**

Standard

Output](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch12-06-writing-to-stderr-instead-of-stdout.html){tabindex="0"}

## 27. **13. Functional Language Features: Iterators and**

Closures](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch13-00-functional-features.html){tabindex="0"}

28. 1. **13.1. Closures: Anonymous Functions that Capture Their**

Environment](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch13-01-closures.html){tabindex="0"}

2. **13.2. Processing a Series of Items with**

Iterators](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch13-02-iterators.html){tabindex="0"}

3. **13.3. Improving Our I/O**

Project](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch13-03-improving-our-io-project.html){tabindex="0"}

4. **13.4. Comparing Performance: Loops vs.**

Iterators](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch13-04-performance.html){tabindex="0"}

29. **14. More about Cargo and**

Crates.io](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-00-more-about-cargo.html){tabindex="0"}

30. 1. **14.1. Customizing Builds with Release**

Profiles](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-01-release-profiles.html){tabindex="0"}

2. **14.2. Publishing a Crate to**

Crates.io](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-02-publishing-to-crates-io.html){tabindex="0"}

### 3. **14.3. Cargo**

Workspaces](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-03-cargo-workspaces.html){tabindex="0"}

### 4. **14.4. Installing Binaries from Crates.io with cargo**

install](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-04-installing-binaries.html){tabindex="0"}

### 5. **14.5. Extending Cargo with Custom**

Commands](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch14-05-extending-cargo.html){tabindex="0"}

## 31. **15. Smart**

Pointers](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-00-smart-pointers.html){tabindex="0"}

### 32. 1. **15.1. Using Box to Point to Data on the**

Heap](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-01-box.html){tabindex="0"}

### 2. **15.2. Treating Smart Pointers Like Regular References with**

Deref](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-02-deref.html){tabindex="0"}

### 3. **15.3. Running Code on Cleanup with the Drop**

Trait](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-03-drop.html){tabindex="0"}

### 4. **15.4. Rc<T>, the Reference Counted Smart**

Pointer](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-04-rc.html){tabindex="0"}

### 5. **15.5. RefCell<T> and the Interior Mutability**

Pattern](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-05-interior-mutability.html){tabindex="0"}

### 6. **15.6. Reference Cycles Can Leak**

Memory](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch15-06-reference-cycles.html){tabindex="0"}

## 33. **16. Fearless**

Concurrency](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch16-00-concurrency.html){tabindex="0"}

### 34. 1. **16.1. Using Threads to Run Code**

Simultaneously](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch16-01-threads.html){tabindex="0"}

### 2. **16.2. Using Message Passing to Transfer Data Between**

Threads](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/

ch16-02-message-passing.html){tabindex="0"}

### 3. **16.3. Shared-State**

Concurrency](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch16-03-shared-state.html){tabindex="0"}

### 4. **16.4. Extensible Concurrency with the Send and Sync**

Traits](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch16-04-extensible-concurrency-sync-and-send.html){tabindex="0"}

## 35. **17. Fundamentals of Asynchronous Programming: Async, Await, Futures, and**

Streams](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-00-async-await.html){tabindex="0"}

### 36. 1. **17.1. Futures and the Async**

Syntax](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-01-futures-and-syntax.html){tabindex="0"}

### 2. **17.2. Applying Concurrency with**

Async](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-02-concurrency-with-async.html){tabindex="0"}

### 3. **17.3. Working With Any Number of**

Futures](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-03-more-futures.html){tabindex="0"}

### 4. **17.4. Streams: Futures in**

Sequence](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-04-streams.html){tabindex="0"}

5. **17.5. A Closer Look at the Traits for**

Async](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-05-traits-for-async.html){tabindex="0"}

6. **17.6. Futures, Tasks, and**

Threads](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch17-06-futures-tasks-threads.html){tabindex="0"}

37. **18. Object Oriented Programming Features of**

Rust](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch18-00-oo.html){tabindex="0"}

38. 1. **18.1. Characteristics of Object-Oriented**

Languages](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch18-01-what-is-oo.html){tabindex="0"}

2. **18.2. Using Trait Objects That Allow for Values of  
Different**

Types](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch18-02-trait-objects.html){tabindex="0"}

3. **18.3. Implementing an Object-Oriented Design**

Pattern](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/c

h18-03-oo-design-patterns.html){tabindex="0"}

## 39. **Patterns and**

Matching](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch19-00-patterns.html){tabindex="0"}

### 40. 1. **All the Places Patterns Can Be**

Used](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch19-01-all-the-places-for-patterns.html){tabindex="0"}

#### 2. **Refutability: Whether a Pattern Might Fail to**

Match](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch19-02-refutability.html){tabindex="0"}

#### 3. **Pattern**

Syntax](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch19-03-pattern-syntax.html){tabindex="0"}

## 41. **Advanced**

Features](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-00-advanced-features.html){tabindex="0"}

### 42. 1. **Unsafe**

Rust](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-01-unsafe-rust.html){tabindex="0"}

#### 2. **Advanced**

Traits](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-02-advanced-traits.html){tabindex="0"}

### 3. **20.3. Advanced**

Types](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-03-advanced-types.html){tabindex="0"}

### 4. **20.4. Advanced Functions and**

Closures](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-04-advanced-functions-and-closures.html){tabindex="0"}

### 5. **20.5.**

Macros](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch20-05-macros.html){tabindex="0"}

## 43. **21. Final Project: Building a Multithreaded Web**

Server](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch21-00-final-project-a-web-server.html){tabindex="0"}

### 44. 1. **21.1. Building a Single-Threaded Web**

Server](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch21-01-single-threaded.html){tabindex="0"}

### 2. **21.2. Turning Our Single-Threaded Server into a Multithreaded**

Server](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch21-02-multithreaded.html){tabindex="0"}



### 3. **21.3.** Graceful Shutdown and

Cleanup](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch21-03-graceful-shutdown-and-cleanup.html){tabindex="0"}

### 45. **22.**

Appendix](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-00.html){tabindex="0"}

### 46. 1. **22.1.** A -

Keywords](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-01-keywords.html){tabindex="0"}

### 2. **22.2.** B - Operators and

Symbols](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-02-operators.html){tabindex="0"}

### 3. **22.3.** C - Derivable

Traits](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-03-derivable-traits.html){tabindex="0"}

### 4. **22.4.** D - Useful Development

Tools](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-04-useful-development-tools.html){tabindex="0"}

### 5. **22.5.** E -

Editions](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/

appendix-05-editions.html){tabindex="0"}

## 6. **22.6.** F - Translations of the

Book](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-06-translation.html){tabindex="0"}

## 7. **22.7.** G - How Rust is Made and "Nightly

Rust"](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-07-nightly-rust.html){tabindex="0"}

:::: {#sidebar-resize-handle .sidebar-resize-handle}

::: sidebar-resize-indicator

:::

::::

:::::::::: {#page-wrapper .page-wrapper}

:::::::::: page

::: {#menu-bar-hover-placeholder}

:::

::::: {#menu-bar .menu-bar style="top: 0px;"}  
::: left-buttons

- Auto
- Light
- Rust
- Coal

- Navy

- Ayu

:::

# The Rust Programming Language {#the-rust-programming-language .menu-title}

::: right-buttons

[(file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/print.html "Print this book"){aria-label="Print this book"}]

[(https://github.com/rust-lang/book "Git repository"){aria-label="Git repository"}]

:::

:::::

::::: {#search-wrapper .hidden}

:::: {#searchresults-outer .searchresults-outer .hidden}

::: {#searchresults-header .searchresults-header}

:::

:::::

:::::

::::: {#content .content}

::: {role="main"}

# [The Rust Programming

Language](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/index.html#the-rust-programming-language){.header}

{#the-rust-programming-language}

\*by Steve Klabnik, Carol Nichols, and Chris Krycho, with contributions from the Rust Community\*

This version of the text assumes you're using Rust 1.85.0 (released 2025-02-17) or later with `edition = "2024"` in the Cargo.toml file of all projects to configure them to use Rust 2024 edition idioms.

See the ["Installation" section of Chapter

1](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/ch01-01-installation.html)

to install or update Rust.

The HTML format is available online at

<<https://doc.rust-lang.org/stable/book/>> and offline with installations of Rust made with `rustup`; run `rustup doc --book` to open.

Several community

[translations](file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/appendix-06-translation.html)

are also available.

This text is available in [paperback and ebook format from No Starch Press](<https://nostarch.com/rust-programming-language-2nd-edition>).

> \*\* Want a more interactive learning experience? Try out a different  
> version of the Rust Book, featuring: quizzes, highlighting,  
> visualizations, and more\*\*:

> [https://rust-book.cs.brown.edu](https://rust-book.cs.brown.edu/)

:::

[(file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/forewor  
d.html "Next chapter")]{.mobile-nav-chapters  
.next rel="next prefetch" aria-label="Next chapter"  
aria-keyshortcuts="Right"}

::: {style="clear: both"}

:::

:::::

:::::::::::

[(file:///home/cinder/Documents/K\_Knowledge\_Base/Rust-Lang-Docs/book/book/forewor  
d.html "Next chapter")]{.nav-chapters  
.next rel="next prefetch" aria-label="Next chapter"  
aria-keyshortcuts="Right"}

:::::::::::

:::::::::::