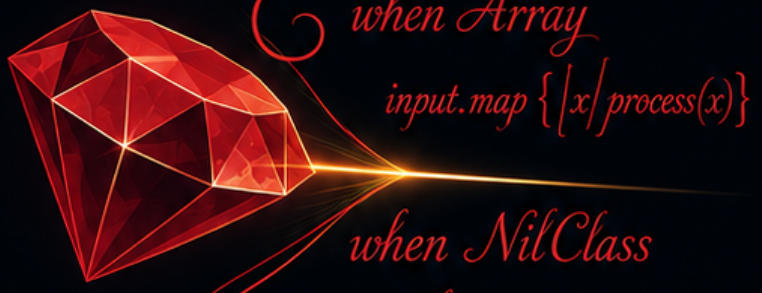
FROM RUBY TO RUST

A Ruby Programmer's Guide to Learning Rust

```
def process(input)
  case input
  when String
    input.strip
  when Array
    input.map {|x| process(x)}
  when NilClass
    nil
  else
    input.to_s
  end
end
```

```
fn process(input: Input) -> Option<String> {
    match input {
        Input::Text(s) => {
            Some(s.trim().to_string())
        }
        Input::List(items) => {
            Some(
                items.into_iter()
                .filter_map(process)
                .collect::<Vec<_>>()
                .join("")
            )
        }
        Input::None => None,
        Input::Other(val) => {
            Some(val.to_string())
        }
    }
}
```



Joel Bryan Juliano

From Ruby to Rust

A Ruby Programmer's Guide to Learning Rust

Joel Bryan Juliano

This book is available at https://leanpub.com/rb2rust

This version was published on 2026-05-29

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Preface

I started writing Ruby in 2010. For over a decade, it was my go-to language — the one I reached for when I needed to think through a problem, build a prototype, or ship a feature. Ruby's slogan is "A Programmer's Best Friend," and I believe it.

In 2018, I added Go to my toolkit, and the Ruby-to-Go mappings I discovered became my first book. During the pandemic, stuck at home like everyone else, I started reading about Rust — its promise of memory safety without a garbage collector, its reputation for catching bugs at compile time, its streak as the most loved language on Stack Overflow. I was hooked before I wrote a single line. I started taking notes. Those notes became a folder. That folder sat on my hard drive, then on Leanpub as an unpublished draft, for years. The method was already proven — teach Rust the same way I'd taught Go, by mapping every concept back to Ruby — but Rust is deeper than Go, and I wanted to get it right.

The first thing I did when learning Rust was try to relate everything back to Ruby. How do I define a class? (You don't — you use a struct with an `impl` block.) How do I handle exceptions? (You don't — you use `Result` and the `?` operator.) How do I share data between threads? (Carefully — the borrow checker enforces the rules.) Every answer was a small revelation about how Rust thinks differently.

This book teaches Rust by mapping every concept back to Ruby. If you know how instance variables work in Ruby, you'll learn how structs work in Rust. If you know `require`, you'll learn `use` and `mod`. If you know `begin/rescue`, you'll learn `match` on

`Result`. No abstract explanations – just the Ruby you know, translated to the Rust you want to learn.

Rust has been the most loved language on Stack Overflow's developer survey for years running. It powers Firefox's rendering engine, parts of the Linux kernel, Discord's backend, Cloudflare's edge workers, and Dropbox's storage systems. When performance matters and correctness is non-negotiable, teams reach for Rust.

But more importantly: Rust will make you a better programmer. Learning to think about ownership, to let the compiler verify your concurrency, to use algebraic data types for domain modeling – these patterns will change how you write code in any language, including Ruby.

This book doesn't ask you to take notes or memorize syntax. Just read, type the examples, and let your Ruby knowledge do the heavy lifting.

Ruby is still my favorite language for getting ideas into code fast. This book is not about leaving Ruby behind – it's about adding Rust to your toolkit for the problems where performance and correctness matter most.

Let's get started.

Introduction

Rust is not an object-oriented language. No classes, no inheritance, no garbage collector, no `nil`. If you're coming from Ruby, that sounds like a step backward. But Rust's approach — structs with `impl` blocks instead of classes, traits instead of inheritance, `Option` instead of `nil`, and the borrow checker instead of GC — produces code that is blazingly fast, memory-safe, and free of entire categories of bugs before your code ever runs.

This book is built on one idea: the fastest way for a Rubyist to learn Rust is to map every Rust concept back to Ruby. Each chapter starts with Ruby code you already understand, then shows the Rust equivalent. You're not learning from scratch — you're translating.

Who This Book Is For

You should know Ruby. You don't need to be an expert — if you can read a Ruby class, understand instance variables, and follow a `begin/rescue` block, you have what you need.

You don't need to know Rust at all. Chapter 1 starts from zero.

You don't need to know C or C++. Rust's ownership system will be new to you, and that's fine — this book devotes two full chapters to it.

How to Use This Book

Read in order. Each chapter builds on the last. Ownership (Chapter 4) underpins everything that follows. Skip around and you'll miss the foundation.

Type the examples. Every code sample is complete — copy, paste, and run with `cargo run`. Then modify it. Remove a `&` and watch the borrow checker complain. Try to mutate a value after moving it. The exercises at the end of each section give you specific things to try.

Read the compiler. Rust's error messages are world-class — they explain what went wrong, why, and often suggest the exact fix. When the compiler complains, read the message carefully. It's teaching you.

Read the Key Points. Each major section has a T> callout box summarizing the one thing to remember. If you're skimming or reviewing, these are your landmarks.

Use the exercises. Every chapter ends with hands-on coding challenges. They're not optional — Rust is a language you learn by typing, not by reading. The compiler is your teacher; the exercises are the curriculum.

What This Book Is Not

This is not a Rust reference. It won't cover every standard library module or every language feature. It covers the concepts a Rubyist needs to become productive in Rust — the things you'll actually use day to day.

If you want depth on specific topics after finishing this book, The Rust Book, Rust by Example, and Rustlings are excellent next steps.

A Note on Editions

This book targets Rust Edition 2024, the current stable edition. All examples have been tested with Rust 1.85+. If you're using an older version, most examples will still work — check `rustc --version` and upgrade with `rustup update` if needed.

About the Author

I've been writing software for over 20 years, from building a Java antivirus engine at VIPRE Security to scaling payment systems at DAZN to 10M+ daily transactions. I started programming as a kid and never stopped. I wrote *From Ruby to Golang* in 2018 and published it in 2019. It reached the top 10 on Amazon and peaked at #3 — proof that the Ruby-to-X method works. That success is why this book exists. I discovered Rust during the pandemic and spent two years learning it before I felt ready to write this book. Rust is a language I believe every backend engineer should know. I live in Amsterdam with my family, where I write, build open-source tools, and work as a Senior Software Engineer.

Chapter 1: Getting Started

Every Rust program starts the same way: a `main` function inside a Cargo project. If you can create a project and write `fn main()`, you're in. Let's get Rust installed and your first program running.

Installing Rust

The recommended way to install Rust is via `rustup` — Rust's toolchain manager. It's like `rbenv` or `rvm` for Ruby, but it's the official solution and comes with the compiler, package manager, and documentation:

```
$ curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

After installation, verify everything is there:

```
$ rustc --version
rustc 1.85.0 (4d91de4e4 2025-02-17)

$ cargo --version
cargo 1.85.0 (d73d7c33d 2025-02-04)
```

`rustc` is the compiler — you'll rarely call it directly. `cargo` is the build tool and package manager — like Bundler and Rake rolled into one.

Creating a New Project

Ruby projects start with a `.rb` file. Rust projects start with `cargo new`:

```
$ cargo new hello_world
     Created binary (application) `hello_world` package
```

This creates:

```
hello_world/
  Cargo.toml
  src/
    main.rs
```

`Cargo.toml` is your Gemfile — it declares dependencies, project metadata, and build settings. `src/main.rs` is your entry point.

Here's what `Cargo.toml` looks like:

```
[package]
name = "hello_world"
version = "0.1.0"
edition = "2024"

[dependencies]
```

And `src/main.rs` already contains a working program:

```rust
fn main() {
    println!("Hello, world!");
}
```

Run it with `cargo run`:

```
$ cargo run
   Compiling hello_world v0.1.0
    Finished dev [unoptimized + debuginfo] target(s)
     Running `target/debug/hello_world`
Hello, world!
```

`cargo run` compiles and runs in one step. Use `cargo build` to compile without running, or `cargo check` to verify it compiles without producing a binary — useful for quick feedback in your editor.

The Main Function

Key Point: `fn main()` is where Rust programs start. It takes no arguments, returns nothing, and every executable must have exactly one.

Every Rust executable needs a `main` function. It's the entry point:

```rust
fn main() {
    // Your program starts here
}
```

Unlike Ruby, where code at the top level executes immediately, Rust requires everything inside `fn main()` (or another function). No top-level `puts` — everything is scoped.

println! — Rust's puts

`println!` works like Ruby's `puts` — it prints a line with a newline:

```
println!("Hello World");
```

The `!` means `println` is a macro, not a function. Macros generate code at compile time. For now, you can use `println!` exactly like `puts` — the `!` is just syntax you'll type.

Semicolons end statements in Rust. They're required — unlike Ruby, where they're optional.

To print without a newline, use `print!`:

```
print!("Loading");
print!("...");
println!(" done!");
// Loading... done!
```

String Interpolation with format!

Ruby's string interpolation uses `#{}`:

```
name = "World"
puts "Hello, #{name}!"
```

Rust's `println!` uses `{}` placeholders:

```rust
let name = "World";
println!("Hello, {}!", name);
```

Multiple values are positional:

```rust
let x = 10;
let y = 20;
println!("x = {}, y = {}, sum = {}", x, y, x + y);
// x = 10, y = 20, sum = 30
```

For debug output, use `{:?}` — this works like Ruby's `inspect`:

```rust
let numbers = vec![1, 2, 3];
println!("numbers = {:?}", numbers);
// numbers = [1, 2, 3]
```

`format!` works like `sprintf` in Ruby — it returns a `String` instead of printing:

```rust
let name = "World";
let greeting = format!("Hello, {}!", name);
println!("{}", greeting);
```

Functions

Key Point: Rust uses functions for everything. No classes, no methods on objects — just `fn`. If you come from Ruby, this is the biggest structural shift, but `impl` blocks (Chapter 3) bring back the familiar dot-call pattern.

Defining Functions

A function starts with `fn`, then the name, then parameters, an optional return type, and a body:

```rust
fn greet() {
    println!("Hello, World!");
}
```

Parameters

Parameters require explicit types — unlike Ruby, Rust won't guess:

```rust
fn greet(name: &str) {
    println!("Hello, {}!", name);
}

greet("World");
```

`&str` is a string slice — Rust's closest equivalent to Ruby's `String`. We'll cover strings in depth in Chapter 9. For now, think of `&str` as "a reference to a string."

Multiple parameters separate with commas. Each needs its own type:

```rust
fn greet(greeting: &str, name: &str) {
    println!("{}, {}!", greeting, name);
}

greet("Hello", "World");
```

Return Values

Return types come after the arrow `->`. In Rust, the last expression without a semicolon is the return value — no `return` keyword needed:

```rust
fn greet(greeting: &str, name: &str) -> String {
    format!("{}, {}!", greeting, name)
}

fn main() {
    let message = greet("Hello", "World");
    println!("{}", message);
}
```

The absence of a semicolon on `format!(...)` means "return this value." This is Rust's expression-oriented nature — nearly everything produces a value.

Use `return` for early returns:

```rust
fn divide(a: f64, b: f64) -> Option<f64> {
    if b == 0.0 {
        return None;
    }
    Some(a / b)
}
```

`Option` is Rust's replacement for `nil` — it's either `Some(value)` or `None`, and the compiler forces you to handle both cases. We'll cover `Option` in depth in Chapter 4.

Cargo Commands Cheat Sheet

Command	What It Does
`cargo new <name>`	Create a new project
`cargo build`	Compile the project
`cargo run`	Compile and run
`cargo check`	Check if it compiles (faster, no binary)
`cargo test`	Run tests
`cargo fmt`	Format your code
`cargo clippy`	Run the linter

Chapter Exercises

1. Create a new Cargo project called `greeter`. Write a `main` function that prints your name, your favorite programming language, and the current date. Use `println!` with `{}` placeholders for all three.

2. Write a function `greet` that takes a `name: &str` and returns a greeting `String`. Call it from `main` and print the result. Then modify it to return a `Result<String, String>` — return `Err("name is empty".to_string())` if the name is empty. Handle the error in `main` with a `match` (we'll cover this properly in Chapter 4 — just follow the pattern for now).

3. Create a second function `add` that takes two `i32` integers and returns their sum. Call it from `main`. Then add a third function `div` that returns `Option<i32>` — `None` if the divisor is zero. Print the results of both with `{:?}` for the `Option`.

Summary

You can now create, build, and run Rust programs. `cargo new` scaffolds your project. `fn main()` is your entry point. `println!` and `format!` replace `puts` and string interpolation. Functions use `fn`, explicit parameter types, and the `->` arrow for return types.

These building blocks never change. Come back here when you need a refresher.

In the next chapter, you'll learn Rust's variable system — `let`, `mut`, shadowing, and how Rust's type inference differs from Ruby's dynamic typing.

Chapter 2: Variables, Types, and Mutability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

let — Variable Binding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Type Inference and Explicit Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Scalar Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Integers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Floats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Booleans

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Characters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Constants

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Shadowing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Compound Types: Tuples and Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Tuples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 3: Structs, Methods, and impl Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Defining a Struct

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Creating Instances

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Accessing Fields

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Struct Update Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Tuple Structs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Unit Structs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

impl Blocks — Adding Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Associated Functions (Constructors)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Multiple impl Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Debug Trait

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ruby Class vs Rust Struct: Side by Side

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 4: Enums and Pattern Matching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Defining an Enum

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

match — Rust's case/when

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Matching Variants with Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

if let — Concise Single-Pattern Matching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Option — The Billion-Dollar Null Replacement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common Option Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Result — Handling Fallible Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Pattern Matching Power Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Matching Ranges

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Matching with Guards

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

@ Bindings

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 5: Ownership and Borrowing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Problem Ownership Solves

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Three Ownership Rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Move Semantics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Copy Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Clone — Explicit Deep Copy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ownership and Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Borrowing and References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Mutable References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Borrowing Rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Multiple Immutable References — OK

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Mutable + Immutable — ERROR

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

One Mutable — OK

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Dangling References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Slices — References to a Portion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ownership in Structs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ruby vs Rust: Memory Management Side by Side

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 6: Collections and Iterators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Vec — Rust's Array

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Creating Vecs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Accessing Elements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Modifying Vecs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Iterating

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

HashMap — Rust's Hash

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common HashMap Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Building a HashMap from Tuples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Iterators — Rust's Enumerable

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Creating Iterators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Iterator Adapters (the map/select/reduce equivalents)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Collecting Results

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Collecting into Result and Option

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Closures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Real-World Iterator Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 7: Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Result — Errors as Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The ? Operator — Propagate Errors Up

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

unwrap and expect

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Handling Different Error Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Box — Catch-All

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Custom Error Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Popular Crates: thiserror and anyhow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

panic! — When Recovery Is Impossible

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Error Handling: Ruby vs Rust

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 8: Traits and Generics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Defining a Trait

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Default Trait Implementations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Trait Bounds and impl Trait

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Generics — Type Parameters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Generic Structs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Generics in impl Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Derive Macros — Auto-Generating Trait Implementations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Trait Objects — Dynamic Dispatch

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common Standard Library Traits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Display — User-Facing Formatting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

From and Into — Type Conversions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Iterator

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Traits vs Ruby Modules: Side by Side

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 9: Strings and Text

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

String and &str — Owned vs Borrowed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

String — Owned, Mutable

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

&str — Borrowed, Immutable

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Converting Between String and &str

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

String Concatenation and format!

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Indexing into Strings

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common String Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Splitting and Joining

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Searching and Replacing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Trimming and Case

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Parsing Strings into Numbers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Building Strings Efficiently

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Bytes, Chars, and Grapheme Clusters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Regular Expressions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Raw String Literals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ruby vs Rust: String Operations Cheat Sheet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 10: Lifetimes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Why Lifetimes Exist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Lifetime Annotation Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

When the Compiler Needs Lifetime Annotations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Lifetime Elision Rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Lifetimes in Structs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Static Lifetime

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Lifetime Annotations in Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common Lifetime Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ruby vs Rust: Lifetimes are Not in Ruby

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 11: Concurrency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Spawning Threads

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Moving Data into Threads

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Message Passing with Channels

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Multiple Producers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Shared State with Arc and Mutex

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Send and Sync — The Safety Traits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Rayon Crate — Parallel Iterators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Concurrency Patterns: Ruby vs Rust

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Choosing a Concurrency Approach

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 12: Modules, Crates, and Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Packages and Crates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Modules — Organizing Code with mod

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

The Module File System Rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Visibility with pub

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

use — Bringing Names into Scope

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

External Crates and Cargo.toml

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Cargo Workspaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Unit Tests (inline)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Assertion Macros

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Integration Tests (in tests/ directory)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Test Organization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Try It Yourself

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Common Project Layout

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Ruby vs Rust: Project Structure Side by Side

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Appendix A: The Rust Toolchain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

rustup — The Toolchain Manager

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Cargo Commands Reference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Editor Setup

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

VS Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Other Editors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Clippy — The Linter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

rustfmt — The Formatter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Useful Crates by Category

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Rust Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

CI Configuration (GitHub Actions)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Next Steps After This Book

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Appendix B: Exercise Answers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 1

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 2

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 4

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 5

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 6

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 7

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 8

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 9

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 10

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 11

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Chapter 12

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Glossary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Acknowledgements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.

Credits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2rust.