

Rust Learning Roadmap

Beginner Level

1. Hello, World!

- **Goal:** Get familiar with Rust's syntax and basic setup.
- **Tasks:** Install Rust, write a simple program, and compile it.

2. Basic Calculator

- **Goal:** Learn about basic data types, arithmetic operations, and user input.
- **Tasks:** Create a program that takes two numbers and an operator (+, -, *, /) as input and prints the result.

3. Guessing Game

- **Goal:** Understand control flow, loops, and conditional statements.
- **Tasks:** Create a game where the program randomly selects a number and the user tries to guess it.

4. Temperature Converter

- **Goal:** Learn about functions and modules.
- **Tasks:** Write functions to convert temperatures between Celsius and Fahrenheit and use modules to organize the code.

5. Simple To-Do List

- **Goal:** Work with vectors, enums, and basic file I/O.
- **Tasks:** Create a command-line to-do list application that can add, remove, and list tasks.

Lower Intermediate Level

6. Unit Converter

- **Goal:** Work with enums and pattern matching.

- **Tasks:** Create a program that converts units (e.g., length, weight) between different measurement systems.

7. File Reader

- **Goal:** Understand file I/O operations in more depth.
- **Tasks:** Write a program that reads a text file and prints its contents to the console.

8. Basic HTTP Server

- **Goal:** Learn about TCP/IP and basic networking.
- **Tasks:** Create a simple HTTP server that responds with a "Hello, World!" message.

Intermediate Level

9. Web Scraper

- **Goal:** Understand error handling, crates (libraries), and basic networking.
- **Tasks:** Write a program that fetches and parses HTML from a website using crates like `request` and `scraper`.

10. Multithreaded File Search

- **Goal:** Learn about concurrency and threading in Rust.
- **Tasks:** Create a tool to search for a text pattern in files using multiple threads to improve performance.

11. REST API

- **Goal:** Work with web frameworks and JSON.
- **Tasks:** Develop a simple REST API using `Rocket` or `Actix` that handles CRUD operations and responds with JSON.

Upper Intermediate Level

12. Simple Command-Line Chat Application

- **Goal:** Implement basic networking and understand client-server architecture.
- **Tasks:** Build a command-line chat application where multiple clients can connect to a server and send messages to each other.

13. Interactive CLI Tool

- **Goal:** Learn about creating user-friendly command-line interfaces.

- **Tasks:** Create an interactive command-line tool with menus and options using a crate like `clap` or `structopt`.

14. Data Visualization Tool

- **Goal:** Work with external APIs and graphical libraries.
- **Tasks:** Fetch data from a public API and visualize it using a library like `plotters`.

Advanced Level

15. Game Development

- **Goal:** Explore graphics programming and game loops.
- **Tasks:** Create a simple game like Snake or Tetris using a library like `ggez` or `amethyst`.

16. Compiler/Interpreter

- **Goal:** Dive into systems programming and low-level concepts.
- **Tasks:** Write a basic interpreter or compiler for a simple language to understand parsing and code generation.

17. Blockchain Implementation

- **Goal:** Learn about cryptography, distributed systems, and advanced data structures.
- **Tasks:** Implement a basic blockchain with features like proof-of-work, transaction validation, and peer-to-peer networking.

18. Operating System Development

- **Goal:** Understand OS fundamentals and low-level hardware interaction.
- **Tasks:** Start a simple operating system kernel using Rust, focusing on booting, memory management, and basic I/O.

Additional Resources

- **Books:** *The Rust Programming Language* (commonly known as "The Book"), *Programming Rust*, *Rust by Example*.
- **Online Courses:** Udemy, Coursera, or Pluralsight courses on Rust.
- **Practice Platforms:** Exercism, LeetCode, and Codewars for coding challenges.
- **Community:** Join the Rust community on Reddit, Discord, or the Rust User Forums for support and networking.