

Rust Coding Tutorial

Sungjin Kim

August 31, 2025

Contents

1	Getting Started with Cargo	5
1.1	Understanding <code>cargo run</code> and Project Structure in Rust	5
1.1.1	<code>cargo run</code>	5
1.1.2	<code>cargo run --example <name></code>	5
1.1.3	How to Create a <code>tutorials</code> Folder	5

Chapter 1

Getting Started with Cargo

1.1 Understanding `cargo run` and Project Structure in Rust

This guide explains how Cargo runs different parts of your Rust project and the conventional way to structure examples and tutorials.

1.1.1 `cargo run`

The standard `cargo run` command is used to compile and run the main binary of your project.

- **For a Binary Crate:** If your project is an executable program, `cargo run` looks for a file named `src/main.rs`. It compiles this file and its dependencies and then runs the resulting program.
- **For a Library Crate:** If your project is a library, it won't have a `src/main.rs` file. Running `cargo run` by itself will result in an error because Cargo doesn't know what to run.

1.1.2 `cargo run --example <name>`

This command is used to run specific example code.

- It looks inside the `examples/` directory for a file named `<name>.rs`.
- It compiles that file as a small, separate program that can use the functions and structs from your main library code (in `src/`).
- This is the standard way in Rust to provide runnable examples that demonstrate your library's features.

For example, to run `examples/tree_visualization.rs`, you would use:

```
cargo run --example tree_visualization
```

1.1.3 How to Create a `tutorials` Folder

Cargo relies on specific folder names like `src`, `examples`, `tests`, and `benches`. A folder named `tutorials` is not a standard convention that Cargo has a special command for.

The best and most idiomatic way to create tutorials is to **treat them as examples**.

You can structure your project like this, placing your tutorial files inside the `examples` directory with descriptive names:

```
my_project/
├── examples/
│   ├── 01_basic_setup.rs
│   ├── 02_advanced_usage.rs
│   └── 03_a_full_tutorial.rs
├── src/
│   └── lib.rs
└── Cargo.toml
```

You would then run each tutorial just like any other example:

```
cargo run --example 01_basic_setup  
cargo run --example 02_advanced_usage  
cargo run --example 03_a_full_tutorial
```

This approach works seamlessly with Cargo's built-in tooling and is the standard practice within the Rust community.