

Template; Rust Programming Language

Gael Zarco

March 18, 2025

A sample template for note-taking in Rust.

1 Introduction and `main` function

Listing 1: Example Program

```
1 fn main() {  
2     println!("Hello, World!");  
3 }
```

Outputs "Hello World!" to the terminal.

The **Rust Programming Language** is a programming language with an emphasis on memory-safety and speed.

- Utilizes a paradigm known as **Ownership**.
- Memory managed by the **Borrow Checker**.

Always contains a `main` function.

1.1 Subection

if program !contain main \rightarrow error.

Listing 2: `main` Function Syntax

```
1 fn main(..args) {  
2     statement(s)  
3 }
```

2 Borrowing and Ownership

Rust's ownership system ensures memory safety without a garbage collector by enforcing strict rules on how values are accessed and modified.

- **Ownership:** Every value in Rust has a single owner, and when the owner goes out of scope, the value is dropped.
- **Borrowing:** Allows references to use a value without taking ownership, preventing multiple mutable references at the same time.
- **Mutable vs Immutable Borrowing:** Multiple immutable references `&T` are allowed, but only one mutable reference `&mut T` can exist at a time.
- **Lifetimes:** Ensure borrowed references do not outlive the data they point to, preventing dangling references.

3 Summary

Rust is a powerful programming language with an emphasis on speed and memory safety.