Programming 101: Introduction to Rust

Welcome to Programming 101

This document is designed to introduce you to basic programming concepts using Rust. It covers fundamental topics like variables, data types, and basic commands for navigating and running Rust projects.

What is a Variable?

A **variable** is a way to store information in a program. It's like a labeled box where you can put different values, such as numbers, text, or yes/no answers.

Declaring Variables

When you **declare** a variable, you create it and give it a name. In Rust, you use the `let` keyword to declare variables:

```
    let x: i32 = 5; // Declares an integer variable named x
```

Basic Data Types in Rust

Here are some common data types in Rust:

- **Integer (`i32`)**: A whole number, like 5 or 10.
- **Boolean (`bool`)**: A true or false value.
- **String (`&str`)**: A sequence of characters, like ”Hello, Rust!”.

Navigating the Project Directory

As you work on different tasks in this course, you will need to navigate between multiple directories. Here are some basic terminal commands to help you:

- `cd <directory>`: Change to a specific directory.
- `pwd`: Print the current working directory.
- `cargo run`: Run a Rust program from the current project directory.

Switching Between Projects

The `virtual-robot-maze` repository contains multiple Rust projects. You'll need to switch directories to work on different tasks:

1. **Switch to the basic programming tasks**:
   - Navigate to the `basic-programming` directory:

     ```
     cd ~/workspaces/virtual-robot-maze/basic-programming/basic_syntax
     ```

2. **Switch to the virtual robot tasks**:
   - Navigate to the `src` directory:

     ```
     cd ~/workspaces/virtual-robot-maze/src
     ```

Running Rust Programs

Once you're in the correct project directory, use the following command to run the program:

```
cargo run
```

Basic Workflow Tips

- Always check your current directory with `pwd` before running commands.
- Save your code frequently to avoid losing progress.
- Use `git add`, `git commit`, and `git push` to save your changes to GitHub.

Next Steps

Once you are comfortable with navigating directories and running basic Rust programs, you can start working on the stepping-stone tasks and virtual robot challenges in the repository.