Programming 101: Introduction to Rust with D&D Characters

Welcome to Programming 101

This document introduces basic programming concepts using Rust and Dungeons & Dragons (D&D) characters. You'll learn to declare variables, understand data types, and use functions to interact with a simulated character sheet.

What is a Variable?

A **variable** is a way to store information in a program. Think of it as a labeled box where you can keep different values, like your character's strength or name.

Declaring Variables

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

```
let strength: u8 = 15; // Declares an integer variable named strength
```

Basic Data Types in Rust

Here are common data types used in Rust:

- **Integer (`u8`)**: A whole number (0-255). Use for attributes like strength.
- **String (`&str`)**: Text, like a character's name or class.
- **Boolean (`bool`)**: A true or false value, like whether a character is currently alive.

Navigating the Project Directory

As you work on different tasks, you'll need to navigate between project directories:

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

Switching Between Projects

To access various tasks:

1. **Basic programming tasks** are in `basic-programming/basic_syntax`:

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

2. **D&D character tasks** are in `src`:

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

Interacting with the D&D API

Using Rust, you can query your character's attributes through specific functions:

- `get_character_name()`: Returns a character's name.
- `get_strength()`: Returns the strength score.
- `get_intelligence()`: Returns the intelligence score.

Running Rust Programs

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

```
cargo run
```

Basic Workflow Tips

- Use `pwd` to check your current directory.
- Save your code regularly.
- Use `git add`, `git commit`, and `git push` to save changes to GitHub.

Next Steps

Once you're comfortable navigating directories and running Rust programs, you're ready to start working with the D&D character tasks. Check each task document for detailed instructions.