Functions

Learn to Code with Rust / Section Review

Functions

- A **function** is a sequence of steps. It's a procedure that encapsulates some logic.
- Functions begin with the **fn** keyword and a name in **snake_case**.
- Rust will run the **main** function automatically. Our code manually invokes other functions.

Parameters and Arguments

- Add a pair of parentheses after the function name.
- A **parameter** is a name for an expected input. A parameter has an associated type.
- Separate multiple parameters with a comma and a space.
- An **argument** is the concrete value passed in for a parameter when a function is invoked.

Return Values

- A **return value** is the final output of a function.
- The function's signature must annotate the return value type. Use the **->** arrow syntax.
- For an explicit return, use the **return** keyword followed by the value. Include a semicolon.

Implicit Return Values

- A function will implicitly return the result of the last evaluated line. Omit the semicolon.
- If a function does not explicitly or implicitly return a value, it will return a **unit**, an empty tuple.

Blocks in Functions

- A **block** creates an independent execution environment within a function body.
- The block can implicitly produce a final value much like a function.
- The product of a block can be assigned to a variable.