

Error Handling

Learn to Code with Rust / Section Review

Errors in Rust

- An **error** is an operation that goes wrong.
- A **recoverable error** is one that we can react to and handle in the code.
- An **unrecoverable** error is one that prevents the program from proceeding.
- A **panic** is a runtime error that occurs when the program cannot proceed.
- The **backtrace** is the list of functions that were running at the point that the error occurred.

The **exit** Function

- The **process::exit** function terminates a program.
- The function accepts a **code** parameter that must be an integer.
- A value of 0 indicates the program exited without an error.
- Any number greater than 0 indicates that the program encountered an error.

Standard Error

- The **println!** macro outputs a message to the standard output.
- The **println!** macro outputs a message to standard error.
- **Standard error** is another output stream/channel to send messages to.
- Code can log to different streams (standard output vs. standard error) to segment messages.

The **Result** Enum

- The **Result** enum models an operation that could be successful or erroneous.
- Two example of functions/methods that return **Result** include **stdin().read_line** and **File::open**.
- The **Ok** variant stores the success data.
- The **Err** variant stores the error data.
- The **match** statement forces a pattern for every enum variant.
- The **if let** statement declares conditional logic for a single enum variant.

Propagating Errors

- A common pattern is to define a function that sends a **Result** back to the caller at any point of error.
- The **Err** variant can store different error messages based on what went wrong.
- To **propagate an error** means to send it back up to the caller.
- The caller can then customize how to react to the error.

The ? Operator

- The ? operator either extracts the data from an **Ok** variant and continues running *or* it returns the **Err** variant with the error message.
- Code can method chain the ? in sequence if every method returns a **Result**.
- With an **Option** enum, the ? operator extracts the value from the **Some** variant and continues execution *or* returns the **None** variant.