

Control Flow

Learn to Code with Rust / Section Review

if and else if

- The **if** statement executes a block when its condition evaluates to true.
- The **else if** statement executes a block when its condition evaluates to true and the original **if** statement's condition evaluates to false.

The **else** Keyword

- The **else** statement executes a block if no previous **if** and **else if** statements have evaluated to true.
- The blocks can produce a value and the complete **if/else if/else** construct can be assigned to a variable.

The **match** Keyword

- The **match** keyword compares a dynamic value with all of its possible variations.
- A **pattern** or **arm** defines the logic to execute for one or more variations of the value.
- When an arm is met, Rust evaluates the corresponding block of code.
- Evaluation stops when a pattern match occurs.
- The **underscore** (`_`) is a catch-all that always matches the value.

Iteration

- Iteration is the process of repeating an action over and over again.
- The **loop** keyword declares a block of code that Rust will execute over and over.
- The **break** keyword terminates the iteration. It is usually connected to a conditional.
- The **continue** keyword stops the current loop and begins a new one.

The **while** Loop

- A **while** loop executes as long as a condition is met ("while something is true").
- The block should mutate some state that will affect a future iteration so that the loop can terminate.
- The **break** and **continue** keywords work with a **while** loop.

Recursion

- **Recursion** is when a function calls itself.
- Each recursion represents a separate, independent execution of the function. The caller remains in a pending state.
- A **base case** is a terminating condition that forces the recursion to stop.
- The **base case** involves some conditional that indicates that the end has been reached.