

Conditionals & Logic

Else Statements

Lua `else` statements will execute a code block if every expression in the `if`-based control structure evaluates to `false`. If an `else` is included in a control structure, the `else` must go at the end.

```
if isSunny then
    print("Go outside!")
else
    --This statement executes when every
    expression in
    --a control statement is false.
    print("Stay indoors.")
end
```

Control Structures

A control structure will run different code blocks based on one or more `true` / `false` statements.

```
if isMonday then
    print("It's the beginning of a great
    week.")
elseif isWednesday then
    print('Halfway through the week!')
else
    print("Enjoy your day!")
end
```

If statements

A Lua `if` statement is the first part of an `if`-based control structure. The `if` statement's boolean expression is the first to be evaluated. An `if` statement does not need to be followed by any `else` or `elseif` statements.

```
if weather == "bad" then
    print("Bring a jacket!")
end
```

Elseif Statements

A Lua `elseif` statement allows for additional boolean expressions to be evaluated in an `if`-based control structure. There is no limit to how many `elseif` statements are in a control structure, but `elseif` statements must be after the `if` statement and before a possible `else` statement.

```
if player == "track star" then
    speed = 10
elseif player == "injured" then
    speed = 1
elseif player == "training" then
    speed = 7
else
    speed = 5
end
```

Code block

A code block is a section of code. Control structures can be set up to determine whether or not they execute.

```
if runCodeBlock then
    -- The following indented code is a code
    block:
    print("Part of the code block.")
    print("Also part of the code block.")
end
```

Boolean expressions

Boolean expressions evaluate to either `true` or `false` and are the conditional statements that make up a control structure. You can also include operators and variables in boolean expressions to make your logic more complex.

```
7 > 5 -- Evaluates to: true
```

And operator

The `and` operator is placed between two boolean expressions. It evaluates to `true` when both the left and right sides are `true`. Otherwise the `and` operator evaluates to `false`.

```
true and 5 == 5 -- Evaluates to true
```

or operator

The `or` operator is placed between two boolean expressions and evaluates to `true` if either side is `true`. Otherwise the `or` operator evaluates to `false`.

```
false or 5 == 5 -- Evaluates to: true
```

Not operator

The `not` operator is placed before a boolean expression and evaluates to the inverse of the expression to its right.

```
not ("Hello" == "Hello") -- Evaluates to:  
false
```

Comparison operators

Comparison operators compare two variables and either return `true` or `false`. The comparison operators in Lua are as follows: `<`, `>`, `<=`, `>=`, `==`, `~=`.

```
8 >= 10 -- Evaluates to: false
```

 **Print**  **Share** ▼