Mentorship
Get support and stay on track

Truth tables

Before you advance any further in the lesson, here's the truth tables for logical AND (`&&`) and logical OR (`||`).

&& (AND)

A	B	A && B
`true`	`true`	`true`
`true`	`false`	`false`
`false`	`true`	`false`
`false`	`false`	`false`

|| (OR)

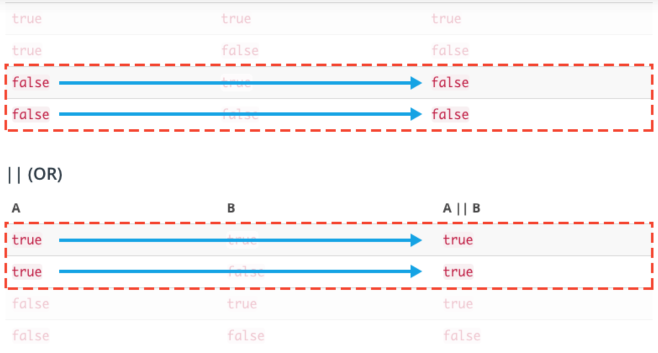
A	B	A || B
`true`	`true`	`true`
`true`	`false`	`true`
`false`	`true`	`true`
`false`	`false`	`false`

Truth tables are used to represent the result of all the possible combinations of inputs in a logical expression. `A` represents the boolean value on the left-side of the expression and `B` represents the boolean value on the right-side of the expression.

Truth tables can be helpful for visualizing the different outcomes from a logical expression. However, do you notice anything peculiar about the truth tables for logical AND and OR?

Short-circuiting

Mentorship
Get support and stay on track



true	true	true
true	false	false
false	true	false
false	false	false

|| (OR)

A	B	A || B
true	true	true
true	false	true
false	true	true
false	false	false

In some scenarios, the value of `B` in logical AND and OR doesn't matter.

In both tables, there are specific scenarios where regardless of the value of `B`, the value of `A` is enough to satisfy the condition.

For example, if you look at `A AND B`, if `A` is *false*, then regardless of the value `B`, the total expression will always evaluate to `false` because both `A` *and* `B` must be `true` in order for the entire expression to be `true`.

This behavior is called **short-circuiting** because it describes the event when later arguments in a logical expression are not considered because the first argument already satisfies the condition.

NEXT