

<

Lesson 3:
Conditionals

☰

📖

📁

🔍

✓ 1. Intro to Conditionals

✓ 2. Quiz: Flowcharts (3-1)

✓ 3. Flowchart to Code

✓ 4. If...Else Statements

✓ 5. Else If Statements

✓ 6. Quiz: Even or Odd (3-2)

✓ 7. Quiz: Musical Groups (3-3)

✓ 8. Quiz: Murder Mystery (3-4)

✓ 9. More Complex Problems

✓ 10. Logical Operators

✓ 11. Logical AND and OR

● 12. Quiz: Checking your Balance (3-5)

● 13. Quiz: Ice Cream (3-6)

● 14. Quiz: What do I Wear? (3-7)

● 15. Advanced Conditionals

● 16. Truthy and Falsy

● 17. Ternary Operator

● 18. Quiz: Navigating the Food Chain (3-8)

● 19. Switch Statement

● 20. Falling-through

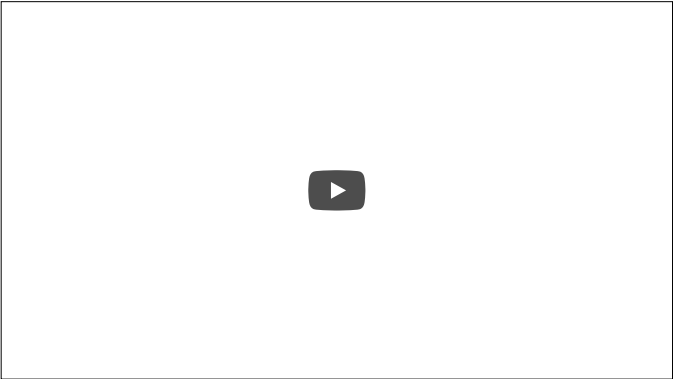
● 21. Quiz: Back to School (3-9)

● 22. Lesson 3 Summary

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

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.