Boolean Logic

Introduction to Boolean Logic

When we learn about if-else statements, comparison and equality operators, and Boolean values, we also need to learn about logic, in particular Boolean logic. Boolean logic is a branch of computer science that uses true and false values to solve complex logical problems. To do this, it uses several logical operators to combine true and false values, much like we use arithmetic operators to combine numeric values.

In this lecture, we will focus on the most basic logical operators: AND, OR, and NOT. Note that Boolean logic is not specific to JavaScript; it applies to all programming languages. Let's now understand how these operators work using an example.

Boolean Variables Example

Consider two Boolean variables, A and B, which can either be true or false. For example, A represents whether Sarah has a driver's license, and B represents whether Sarah has good vision. Both are conditions that can be true or false because Sarah could have no driver's license and no good vision, or she could have both.

The AND Operator

Using the AND operator, expressed as A AND B, we combine these two Boolean variables to represent "Sarah has a driver's license and good vision." To determine the result of this operation, we use a truth table, which shows all possible combinations of A and B and their corresponding results for the AND operation.

The truth table for the AND operator is:

A	B	A AND B
true	true	true
true	false	false
false	true	false
false	false	false

From this table, we see that the AND operation returns true only if both A and B are true. In all other cases, it returns false. This logic extends to more than two operands; for example, A AND B AND C returns true only if all operands are true, otherwise false.

The OR Operator

The OR operator works somewhat oppositely. Using the same example, A OR B represents "Sarah has a driver's license or good vision." The OR operator returns true if at least one of the variables is true. The truth table for OR is:

A	B	A OR B
true	true	true
true	false	true
false	true	true
false	false	false

Thus, the OR operation is true if either A or B is true, and false only if both are false. This also generalizes to multiple operands; the OR operation returns true if at least one operand is true.

The NOT Operator

The NOT operator is simpler because it acts on only one Boolean value and inverts it. If A is true, then NOT A is false; if A is false, then NOT A is true. This operator does not combine multiple values but simply negates a single Boolean value.

Practical Example

Let's consider a practical example using an age variable set to 16. We define two Boolean variables:

- A: age is greater than or equal to 20
- B: age is less than 30

Evaluating these, since age is 16, A is false and B is true.

Now, let's combine these variables using logical operators:

- NOT A: Since A is false, NOT A is true.
- A AND B: False AND true results in false.
- A OR B: False OR true results in true.
- NOT A AND B: True AND true results in true.
- A OR NOT B: False OR false results in false.

Note that the NOT operator has precedence over AND and OR, so values are inverted first before being combined.

Summary

Boolean logic operators AND, OR, and NOT allow us to combine and manipulate true and false values to form complex logical expressions. Understanding their behavior through truth tables and practical examples is essential for programming and logical reasoning.

In the next lecture, we will use these operators in code, which will help solidify your understanding and make these concepts intuitive without needing to refer to truth tables.

Key Takeaways

- Boolean logic uses true and false values to solve complex logical problems.
- The AND operator returns true only if all operands are true.
- The OR operator returns true if at least one operand is true.
- The NOT operator inverts a single Boolean value.