Logical Operators

In the following lesson, you will be introduced to logical operators.

We'll cover the following

Types of Logical Operators

Logical operators are operators that perform logic operations such as the Logical *AND* and Logical *OR*. They take `bool` type operands and yield `bool` type results. Below is a list of the logical operators supported by Dart.

Operator	Name	Use
`!`	Logical NOT	Reverses the logical state of its operand. If a condition is true, then the Logical *NOT* operator will make it false
`||`	Logical OR	If any of the two operands is not false, then the result is true

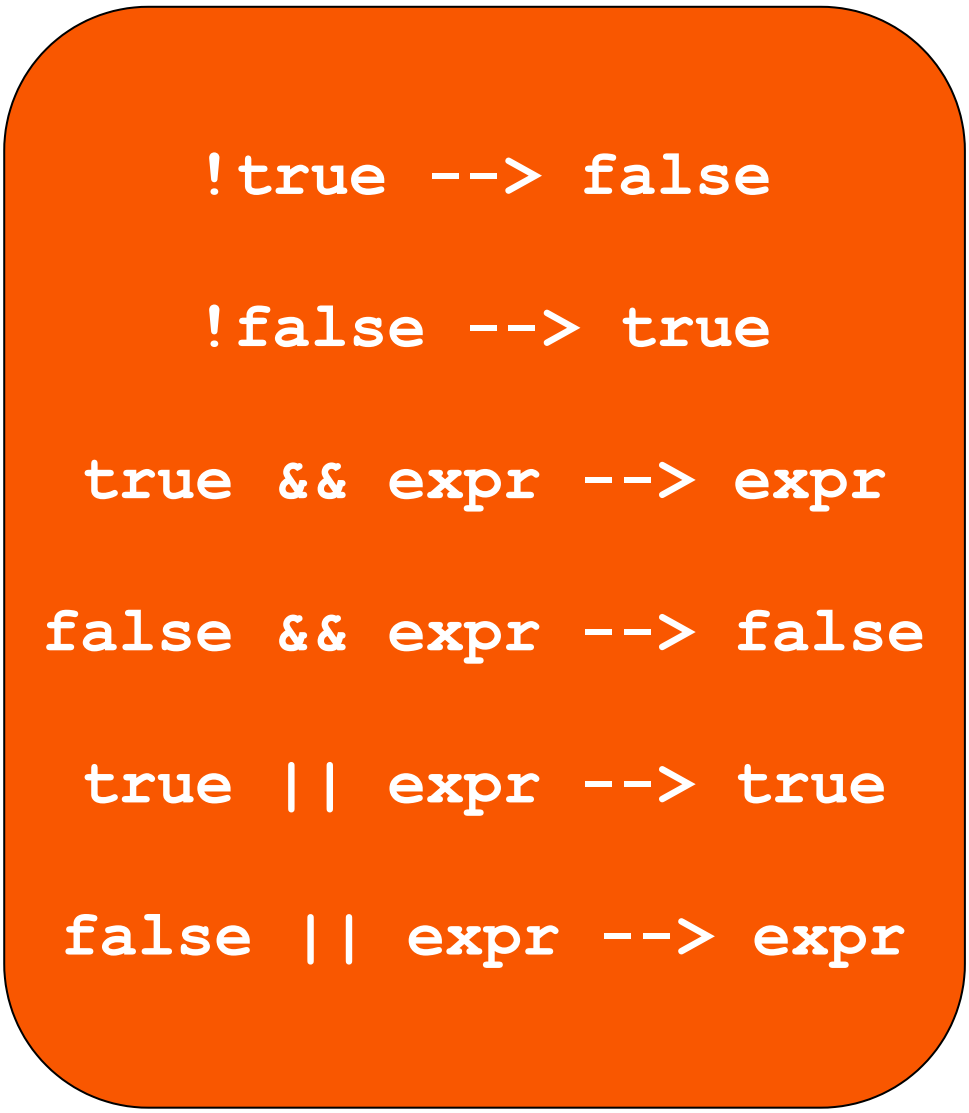
`&&`	Logical AND	If both the operands are not false, then the result is true

> `!` is a unary operator, i.e., it takes one operand.

Follow the Rules

Below, you'll find a list of the reduction rules for logical operators. The list is handy as it will summarize how each operator reduces expressions into their final form.

expr is an arbitrary expression that can be replaced with an operand of type `Boolean`. The operand can be `true` or `false` itself or can be an expression that reduces to `true` or `false.`



Example

Let's now see the above rules in action. For example, our arbitrary expression `expr` will be `A && B` where `A` is `true`, and `B` is `false`.

> Try to figure out what the output would be before pressing **RUN**.

```
main() {
  var A = true;
  var B = false;
  var expr = A && B; //false

  print(!A); // !true --> false
  print(!B); // !false --> true
  print(true || expr); // true || expr --> true
  print(false || expr); // false || expr --> expr
  print(true && expr); // true && expr --> expr
  print(false && expr); // false && expr --> false
}
```

`A && B` reduces to `false` as `B` is `false` and from our list of rules, we know that `false && expr --> false`.

That sums up logical operators. Let's move on to our final type of operators, bitwise and shift operators, in the next lesson.