An Introduction to Operators

In the following lesson, you will be given an overview of the operators Scala provides.

We'll cover the following

Operators are symbols that perform *operations* used for modifying or manipulating data. Manipulating data is an essential part of any programming language, and Scala is no different, providing a rich set of operators for its basic types.

Types of Operators

As far as built-in operators go, Scala provides the following five types.

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators

In the coming lessons, we will be discussing each type of operator individually.

Operators Are Methods

In the previous lesson, you were introduced to methods. Operators are simply methods called using the operator notation we discussed in the previous lesson.

Operators usually follow an infix notation. The infix notation is where the operator is situated between two operands. Operands are the data objects that the operator is performing an operation on.

Operand
Operator
Operand

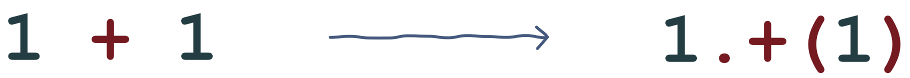
After the operator has performed its specific operation, the output is the result of the operation.

What does this remind you of? A method. Where the operator is our function/method and the operand before the operator is the object the method acts upon. While the operand after the operator is the argument passed to the method.

> In Scala operators are not special language syntax; any method can be an operator. What makes a method an operator is how you use it. When we wrote `string1.indexOf('W')` in the previous lesson, `indexOf` was not an operator. But when we wrote `string1 indexOf 'W'`, `indexOf` was an operator, because we're using it in operator notation.

In the previous lesson, we used the infix notation to call a method. In this lesson, let's try to use an operator using the ordinary method call.

> Note: The infix notation is a much simpler syntax and is the conventional way of using operators.



This code requires the following environment variables to execute:

LANG	C.UTF-8

```
val infix = 1+1
val ordinary = 1.+(1)

// Driver Code
print("Using infix notation: ")
println(infix)

print("Using ordinary method call: ")
println(ordinary)
```

Pretty cool; both variables give us the same output. This example perfectly confirms that operators are just methods with their own syntax.

Let’s now move onto the different types of operators, starting with arithmetic operators in the next lesson.