

VBSCRIPT - OPERATORS

What is an operator?

Simple answer can be given using expression $4 + 5$ is equal to 9. Here, 4 and 5 are called operands and + is called operator. VBScript language supports following types of operators:

- Arithmetic Operators
- Comparison Operators
- Logical or Relational Operators
- Concatenation Operators

The Arithmetic Operators

There are following arithmetic operators supported by VBScript language:

Assume variable A holds 5 and variable B holds 10, then:

[Show Examples](#)

Operator	Description	Example
+	Adds two operands	$A + B$ will give 15
-	Subtracts second operand from the first	$A - B$ will give -5
*	Multiply both operands	$A * B$ will give 50
/	Divide numerator by denominator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	$B \text{ MOD } A$ will give 0
^	Exponentiation Operator	$B ^ A$ will give 100000

To understand these operators in a better way, you can [Try it yourself](#).

The Comparison Operators

There are following comparison operators supported by VBScript language:

Assume variable A holds 10 and variable B holds 20, then:

[Show Examples](#)

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	$A == B$ is False.
<>	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	$A <> B$ is True.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	$A > B$ is False.
<	Checks if the value of left operand is less than the value of right	$A < B$ is

	operand, if yes then condition becomes true.	True.
$>=$	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	$A >= B$ is False.
$<=$	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	$A <= B$ is True.

To understand these operators in a better way, you can [Try it yourself](#).

The Logical Operators

There are following logical operators supported by VBScript language:

Assume variable A holds 10 and variable B holds 0, then:

[Show Examples](#)

Operator	Description	Example
AND	Called Logical AND operator. If both the conditions are True then Expression becomes true.	$a <> 0$ AND $b <> 0$ is False.
OR	Called Logical OR Operator. If any of the two conditions are True then condition becomes true.	$a <> 0$ OR $b <> 0$ is true.
NOT	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	NOT $a <> 0$ OR $b <> 0$ is false.
XOR	Called Logical Exclusion. It is the combination of NOT and OR Operator. If one, and only one, of the expressions evaluates to True, result is True.	$a <> 0$ XOR $b <> 0$ is false.

To understand these operators in a better way, you can [Try it yourself](#).

The Concatenation Operators

There are following Concatenation operators supported by VBScript language:

Assume variable A holds 5 and variable B holds 10 then:

[Show Examples](#)

Operator	Description	Example
+	Adds two Values as Variable Values are Numeric	$A + B$ will give 15
&	Concatenates two Values	$A & B$ will give 510

Assume variable A="Microsoft" and variable B="VBScript", then:

Operator	Description	Example
+	Concatenates two Values	$A + B$ will give MicrosoftVBScript
&	Concatenates two Values	$A & B$ will give MicrosoftVBScript

Note: Concatenation Operators can be used for numbers and strings. The Output depends on the context if the variables hold numeric value or String Value.

To understand these Operators in a better way, you can [Try it yourself.](#)

Loading [MathJax]/jax/output/HTML-CSS/jax.js