**Operators in VB script**

The following table lists the common arithmetic operators available in VBScript:

|  |  |
| --- | --- |
| **Operator** | **Description** |
| **+** | Addition |
| **-** | Subtraction |
| **\*** | Multiplication |
| **/** | Division |
| **\** | Integer Division (divides two numbers and returns an integer result) |
| **Mod** | Modulus (divides two numbers and returns the remainder) |
| **^** | Exponentiation (raises the number to the power of an exponent) |

For example

Dim a  
Dim b  
a = 4  
b = 5  
addition = a + b   
'addition has the value 9  
substract = b - a   
'substract has the value 1  
multiply = a \* b   
'multiply has the value 20  
divide = b / a   
'divide has the value 1.25  
intDivision = b\a   
'intDivision has the value 1  
remainder = b mod a   
'remainder has the value 1  
raise = b ^ a   
'raise now has the value 625

**Data types in VB scripts**

VBScript is a loosely typed programming language. Being a loosely typed programming language, VBScript has only one data type called a **Variant.**It is a special kind of data type, having the capability to store different kinds of information, depending on how it is used. A variant can contain a numeric value or it can also contain string information.   
  
A variant behaves as a number when we use it in a numeric context and as a string when we use it in a string context.

**Variant Subtypes:**Beyond the simple numeric or string classifications, we can further classify variants on the basis of the information they contain. For example, we can have numeric information that represents a date or a time. We have a rich variety of numeric information ranging in size from Boolean values to huge floating-point numbers. These different categories of information are, sometimes, called subtypes of the variant data type. Most of the time, we just put the kind of data we want in a variant, and the variant behaves in a way that is most appropriate for the data it contains.   
  
For the sake of completeness, we have summarized the subtypes of data that a variantcan contain in the following table:

|  |  |
| --- | --- |
| **Subtype** | **Description** |
| **Empty** | Variant is uninitialized. Value is 0 for numeric variables or a zero-length string ("") for string variables. |
| **Null** | Variant intentionally contains no valid data. |
| **Boolean** | Contains either True or False. |
| **Byte** | Contains integer in the range 0 to 255. |
| **Integer** | Contains integer in the range -32,768 to 32,767. |
| **Currency** | -922,337,203,685,477.5808 to 922,337,203,685,477.5807. |
| **Long** | Contains integer in the range -2,147,483,648 to 2,147,483,647. |
| **Single** | Contains a single-precision, floating-point number in the range -3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values. |
| **Double** | Contains a double-precision, floating-point number in the range -1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values. |
| **Date (Time)** | Contains a number that represents a date between January 1, 100 to December 31, 9999. |
| **String** | Contains a variable-length string that can be up to approximately 2 billion characters in length. |
| **Object** | Contains an object. |
| **Error** | Contains an error number. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Description** | **Example** | **Result** |
| + | Adds numbers together.  It also concatenates strings together. | 5+3 | 8 |
| - | Used to subtract numbers.  It can be used to indicate sign of a number. eg: -5 | 5-3 | 2 |
| \* | Used to multiple numbers together | 5\*2 | 10 |
| / | Used to divide numbers.  It returns decimal number. | 5/2 | 2.5 |
| \ | Used to divide numbers.  It returns integer number. | 5\2 | 2 |
| ^ | Raises a number to a power. | 5^2 | 25 |
| & | Used to concatenate strings together | "I"&"AM" | I AM |
| = | Used to assign a value to a variable or compare values. | Num=33 | Num=33 |
| < | Tests if first expression is less than second expression. | Num<55 | true |
| > | Tests if first expression is greater than second expression | Num>55 | false |
| <= | Tests if first expression is less than or equal to second expression | Num<=33 | true |
| >= | Tests if first expression is greater than or equal to second expression. | Num>=22 | false |
| <> | Tests if first expression is not equal to second expression | Num<>31 | true |
| And | Performs a logical conjunction on two expressions or bit-by-bit comparison of two integers. | See if statement |  |
| Or | Performs a logical conjunction on two expressions or bit-by-bit comparison of two integers. | See if statement |  |
| Not | Perform a logical negation on an expression. | not = 22 | true |
| Mod | Returns the reminder of two divided numbers. | 5 mod 2 | 1 |

**VBScript String Concatenation**

Often in our VBScript programs we need to join two or more string together. For string concatenation we have a special operator in VBScript i.e. &. We can use the concatenation operator to join string variables or raw string.

For example

Dim h  
Dim w  
h = "Hello "  
w = "World!"  
hw = h & w