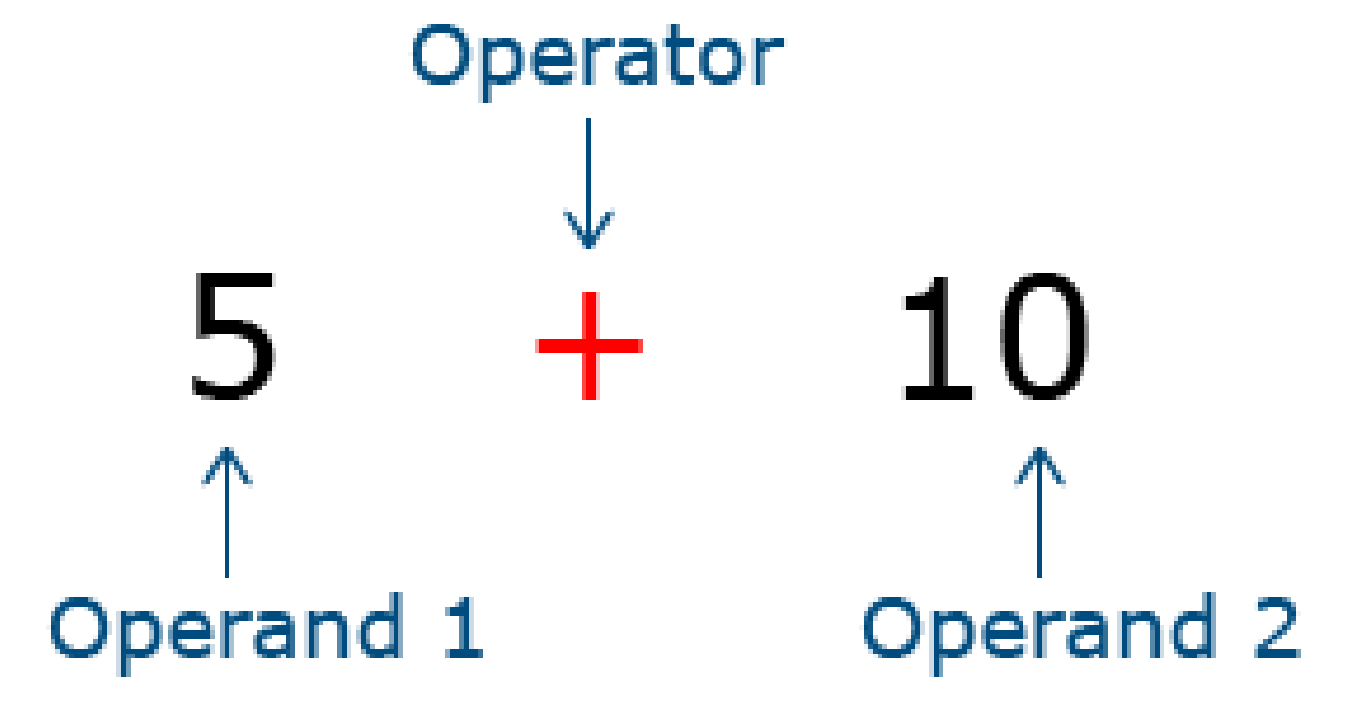
netlink
digital solutions

JavaScript

www.netlink.com

JavaScript : Operators

- In JavaScript, an operator is a special symbol used to perform operations on operands (values and variables).
- This symbol that tells the computer to perform certain mathematical and logical operations.



JavaScript : Operators

There are different types of operators in JavaScript that are used for performing different operations. Some of the JavaScript Operators include:

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Bitwise Operators
- Logical Operators

JavaScript : Arithmetic Operators

Operator	Description	Example
+	Adds two operands	10 + 20 = 30
–	Subtracts the second operand from the first	30 – 20 = 10
/	Divide the numerator by the denominator	20/10 = 2
*	Multiply two operands	5 * 5 = 25
%	Outputs the remainder of an integer division	20 % 10 = 0
++	Increases an integer value by one	var a=20; a++; Now a = 21
—	Decreases an integer value by one	var a=20; a–; Now a = 19

JavaScript : Assignment Operators

Operator	Description	Example
=	Assigns values from the right side operand to the left side operand	20+10 = 30
+=	It adds the right operand to the left operand and assigns the result to the left operand	var a=20; a+=10; Now a = 30
-=	It subtracts the right operand from the left operand and assigns the result to the left operand	var a=30; a-=10; Now a = 20
*=	It multiplies the right operand with the left operand and assigns the result to the left operand	var a=10; a*=20; Now a = 200
/=	It divides the left operand with the right operand and assigns the result to the left operand	var a=10; a/=2; Now a = 5
%=	It takes modulus using two operands and assigns the result to the left operand	var a=10; a%=2; Now a = 0

JavaScript : Logical Operators

Operator	Description	Example
&&	Logical **AND** – If both the operands are non-zero, then the condition becomes true	(10==20 && 20==33) = false
||	Logical **OR** – If any of the two operands are non-zero, then the condition becomes true.	(10==20 || 20==33) = false
!	Logical **NOT** – Reverses the logical state of its operand.	!(10==20) = true

JavaScript : Bitwise Operators

Operator	Description	Example
&	Boolean AND operation on each bit of its integer arguments	(10==20 & 20==33) = false
|	It performs a Boolean OR operation on each bit of its integer arguments	(10==20 | 20==33) = false
^	This operator performs Bitwise XOR operation	(10==20 ^ 20==33) = false
~	It is a unary operator and operates by reversing all the bits in the operand	(~10) = -10
<<	Moves all the bits in its first operand to the left by the number of places specified in the second operand.	(10<<2) = 40
>>	The left operand's value is moved right by the number of bits specified by the right operand.	(10>>2) = 2
>>>	This operator is just like the >> operator, except that the bits shifted in on the left are always zero.	(10>>>2) = 2

JavaScript : Comparison Operators

Operator	Description	Example
==	Checks if two operands are equal or not. If yes, then the condition becomes true.	20==30 = false
===	Finds the identical (equal and of the same type)	10==20 = false
!=	Checks if two operands are equal or not. If the values are not equal, then the condition becomes true	20!=30 = true
!==	It implies that two values are not Identical	20!==20 = false
>	Checks if the value of the left operand is greater than the value of the right operand	30>10 = true
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand	20>=10 = true
<	This Checks if the value of the left operand is less than the value of the right operand	20<10 = false
<=	Checks if the value of the left operand is less than or equal to the value of the right operand	30<=10 = false

Thank you