NIELIT Gorakhpur

COURSE NAME: O level **SUBJECT: WEB DESIGNING AND PUBLISHING**

TOPIC: JavaScript **DATE: 19/05/2020**

JavaScript Code Blocks

We may group JavaScript statements together in code blocks inside the curly brackets {...}. The purpose of code blocks is to define statements to be executed together.

Keywords in JavaScript

There are JavaScript statements which are begin with a **keyword** to specify what JavaScript action to be performed.

Keywords are words which cannot be used for any other purposes like naming variable,identifiers etc

Some Keywords:

Keyword	Description
break	Used to Terminate a switch or a loop
continue	Used to Jump out of a loop and starts at the top
debugger	Used to stop the execution of JavaScript code , and calls the debugging function, if available
do ... while	Used to Execute a block of statements, and repeats it while the condition is true
for	Used to execute a block of statement as long as a condition is true
function	Used to Declare a function
if ... else	Used to execute a block of statements based on the condition
return	Used to exit a function
switch	Used to execute a block of statements based on the condition on different cases
try ... catch	Used for error handling to a block of statements
var	Used to Declare a variable

Operators in JavaScript

1. Basic Assignment Operator

'=' is used as basic assignment operator in JavaScript. A value at the right side is assigned to the variable on the left side. It is used for both string and numeric values.

For Example

var a = 7; will assign the value 7 to variable a
var b = 5; will assign the value 5 to b
var c = a +b ; will assign the value of (a+b) i.e. 7 to variable c

var str = “NIELIT Gorakhpur”; will assign string “NIELIT Gorakhpur” to variable str

2. Arithmetic Operators

JavaScript supports the following Arithmetic operators:

Operator	Purpose of the operator
+	Addition
-	Subtraction
*****	Multiplication
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement
******	Exponentiation

3. Other Assignment Operators

Apart from basic assignment operator =, JavaScript also support combination of Assignment and Arithmetic operator as supported by several other languages. This reduces the code.

Operator	Purpose	Example	Similar to (traditional operation)
+=	**addition assignment**	a += b	a = a + b
-=	Subtraction Assignment	a -= b	a = a – b
***=**	Multiplication Assignment	a *= b	a = a * b
/=	Division Assignment	a /= b	a = a / b
%=	Modulus Assignment	a %= b	a = a % b
****=**	Exponentiation Assignment	a **= b	a = a ** b

4. Comparison Operators

JavaScripts supports standard comparison operators.

Operator	Description
==	Equal to
===	Equal value and equal type
!=	Not equal
!==	Not equal value or not equal type
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

5. Logical Operators

JavaScript provides three Logical operators.

Operator	Description
&&	logical and
||	logical or
!	logical not

6. Type Operators

JavaScript supports two Type Operators to know the type of variable or validate instance of an object.

Operator	Description
typeof	Returns the type of a variable
instanceof	Returns true if an object is an instance of an object type

Example

```
typeof 1   --  'number'
typeof '1' -- 'string'
typeof {name: 'nielit'} -- 'object'
typeof true -- 'boolean'
```

Example

```
var mobile = ["LG", "SONY", "VIVO"];

mobile instanceof Array;         will Returns true
mobile instanceof Object;        will Returns true
mobile instanceof String;        will  Returns false
mobile instanceof Number;        will Returns false
```

7. Concatenation Operators or String Operators

The **+** and **+=** operator may also be used be used to add or concatenate strings. It may also be used with a combination of strings and numbers and in this case output will be a string.

Example

```
var str1 = "NIELIT";
var str22 = "DELHI";
var str3 = str1 + " " + str2;
```

The output value of str3 will be : "NIELIT DELHI"

Example

```
var str4 = "Good ";

var str5 = "Morning";

str4 += str5
```

The output value of str4 will be : "Good Morning"

Example

```
var a = 5 + 6;          -> 11 and is number
var b = "5" + 6;        -> 56 and is a string
var c = "Good" + 9;     -> Good9 and is a string
```

8. Ternary or Conditional Operator

The ternary operator assigns a value to a variable based on the condition.

Syntax

variablename* = (*condition*) ? *value1*:*value2

Example

```
IsMinor= (age < 18) ? "Yes":"No";
```

9. Bitwise Operators

Like various programming languages, JavaScript also supports bit-wise operations. All the numbers in JavaScript are stored as a 64-bit floating point number but the bit-wise operation is performed on a 32-bit binary number.

To perform a bit-operation JavaScript converts the number into a 32-bit signed number, perform the operation and finally convert back the result to a 64-bit number.

Operator	Description	Example	Same as	Result in Binary	Result in Decimal
&	AND	5 & 1	0101 & 0001	0001	1
|	OR	5 | 1	0101 | 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Zero fill left shift	5 << 1	0101 << 1	1010	10

>>	Signed right shift	5 >> 1	0101 >> 1	0010	2
>>>	Zero fill right shift	5 >>> 1	0101 >>> 1	0010	2

Example Bit-wise AND (&) : & is a binary operator i.e. accepts two operands and returns 1 if both the bits are set (i.e 1) and 0 in any other case.

A (=5)	B(=1)	OUTPUT (A & B) (=1)
0	0	0
1	0	0
0	0	0
1	1	1

10. **in operator**

The **in** operator returns true if the property specified is in the given object, otherwise returns false.

Example

var mobile = ["LG", "SONY", "VIVO"];

"LG" in mobile will Returns false as it needs the index number instead of value

0 in mobile will Returns true index value 0
1 in mobile will Returns true index value 1

4 in mobile will Returns false index value 4 which does not exist
"length" in mobile will Returns true as length is an Array property

Assignments

1. What are operators in JavaScripts? Explain them.
2. What is Ternary operator? Explain with example.
3. What are keywords? What are their usages?