NIELIT Gorakhpur

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

TOPIC: JavaScript **DATE: 20/05/2020**

Operator Precedence

Operator precedence is very important when we are evaluating arithmetic expressions. It describes the order in which operations are performed in an arithmetic expression. ***Expression grouping ( )*** has the highest precedence. The table below shows the precedence order from Highest to lowest.

Precedence order Highest to lowest	**Operator**	**Description**	**Example**
1	( )	Expression grouping	(3 + 4)
2	.	Member	student.name
2	[]	Member	student["name"]
2	()	Function call	myFunction()
2	new	Create	new Date()
3	++	Postfix Increment	i++
3	--	Postfix Decrement	i--
4	++	Prefix Increment	++i
4	--	Prefix Decrement	--i
4	!	Logical not	!(x==y)
4	typeof	Type	typeof x
5	**	Exponentiation (ES2016)	5**3
6	*	Multiplication	5*9
6	/	Division	20/ 4
6	%	Division Remainder	20 % 3
7	+	Addition	6 +7
7	-	Subtraction	7 - 6
8	<<	Shift left	a << 3
8	>>	Shift right	a >> 3
8	>>>	Shift right (unsigned)	a >>> 3
9	<	Less than	a<b
9	<=	Less than or equal	a <= b

9	>	Greater than	a>b
9	>=	Greater than or equal	a >= b
9	in	Property in Object	"PI" in Math
9	instanceof	Instance of Object	instanceof Array
10	==	Equal	a == b
10	===	Strict equal	a === b
10	!=	Unequal	a != b
10	!==	Strict unequal	a !== b
11	&	Bitwise AND	a & b
12	^	Bitwise XOR	a ^ b
13	|	Bitwise OR	a | b
14	&&	Logical AND	a && b
15	||	Logical OR	a || b
16	? :	Conditional (ternary)	? "Yes" : "No"
17	+=	Assignment	a += b
17	/=	Assignment	a /= b
17	-=	Assignment	a -= b
17	*=	Assignment	a *= b
17	%=	Assignment	a %= b
17	<<=	Assignment	a <<= b
17	>>=	Assignment	a >>= b
17	>>>=	Assignment	a>>>= b
17	&=	Assignment	a &= b
17	^=	Assignment	a ^=b
17	|=	Assignment	a |= b
18	yield	Pause Function	yield a
19	,	Comma	7, 8

Example

```
var a = 150 + 60 * 4;
```

In the above expression,

- The order of precedence is very important, it can be computed as either (i) 210 *4 or (ii) 150 + 240, i.e. What will be done first addition or multiplication
- As per operator precedence, Multiplication **(*)** has higher **precedence** than addition (**+**). So it is better to always use parentheses:

```
var a = (150 + 60 ) * 4;
```

- When using parentheses, the operations inside the parentheses are computed first.
- When many operations have the same precedence (like addition and subtraction), they are computed from left to right:

```
var a = 150 + 60 - 4;
```

Data Types in JavaScript

Data types are used to specify what kind of data can be stored in the variable and manipulated within a program.

Data types in JavaScript are divided into three main categories:

1) **Primitive (or *primary*) data types**
    a. String, Number, and Boolean
    b. can hold only one value at a time
2) ***Composite* (or *reference*) data types**
    a. Object, Array, and Function
    b. composite data types can hold collections of values and more complex entities
3) ***Special* data types: Undefined and Null**

1. String Data Type

The *string* data type is used to represent textual data i.e. a single character or sequences of characters. String data types hold data within single or double quotes.

Example

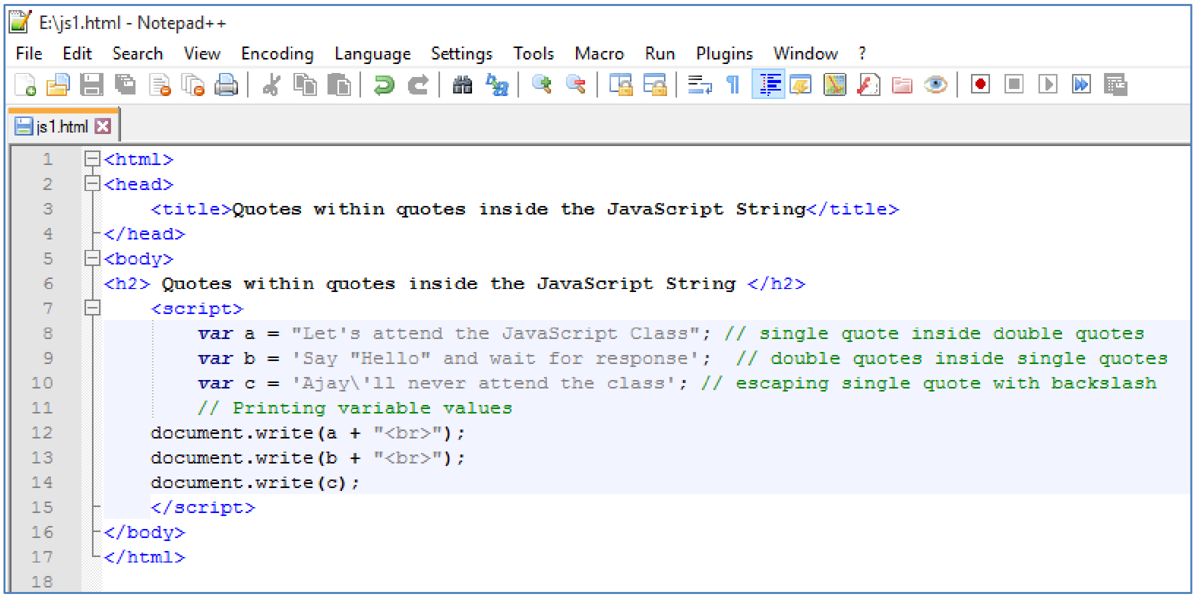
var a = ‘NIELIT’; // uses single enclosing quotes
var b = "NIELIT"; // uses double enclosing quotes

We may also include the quotes within the enclosing quotes inside the string as long as they don't match the enclosing quotes.

Example

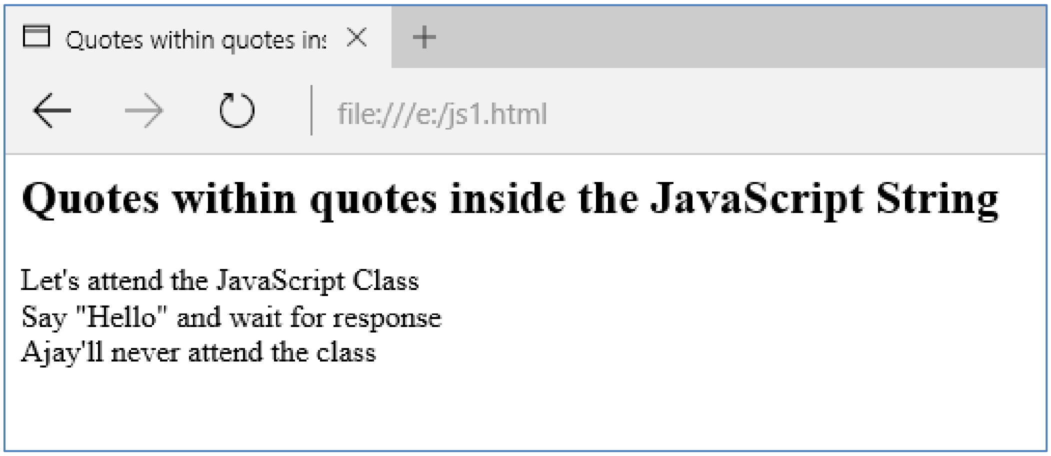
var a = "Let's attend the JavaScript Class"; // single quote inside double quotes
var b = 'Say "Hello" and wait for response'; // double quotes inside single quotes

var c = 'Ajay\'ll never attend the class'; // escaping single quote with backslash


E:\js1.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
js1.html

```
<html>
<head>
    <title>Quotes within quotes inside the JavaScript String</title>
</head>
<body>
<h2> Quotes within quotes inside the JavaScript String </h2>
    <script>
        var a = "Let's attend the JavaScript Class"; // single quote inside double quotes
        var b = 'Say "Hello" and wait for response';  // double quotes inside single quotes
        var c = 'Ajay\'ll never attend the class'; // escaping single quote with backslash
        // Printing variable values
    document.write(a + "<br>");
    document.write(b + "<br>");
    document.write(c);
    </script>
</body>
</html>
```

Output



Quotes within quotes inside the JavaScript String

Let's attend the JavaScript Class
Say "Hello" and wait for response
Ajay'll never attend the class

2. Number Data Type

The *number* data type in JavaScript is used to represent positive or negative numbers with or without decimal place, or numbers written using exponential notation e.g. 1.7e-12 equivalent to 1.7x10^{-12}.

Example

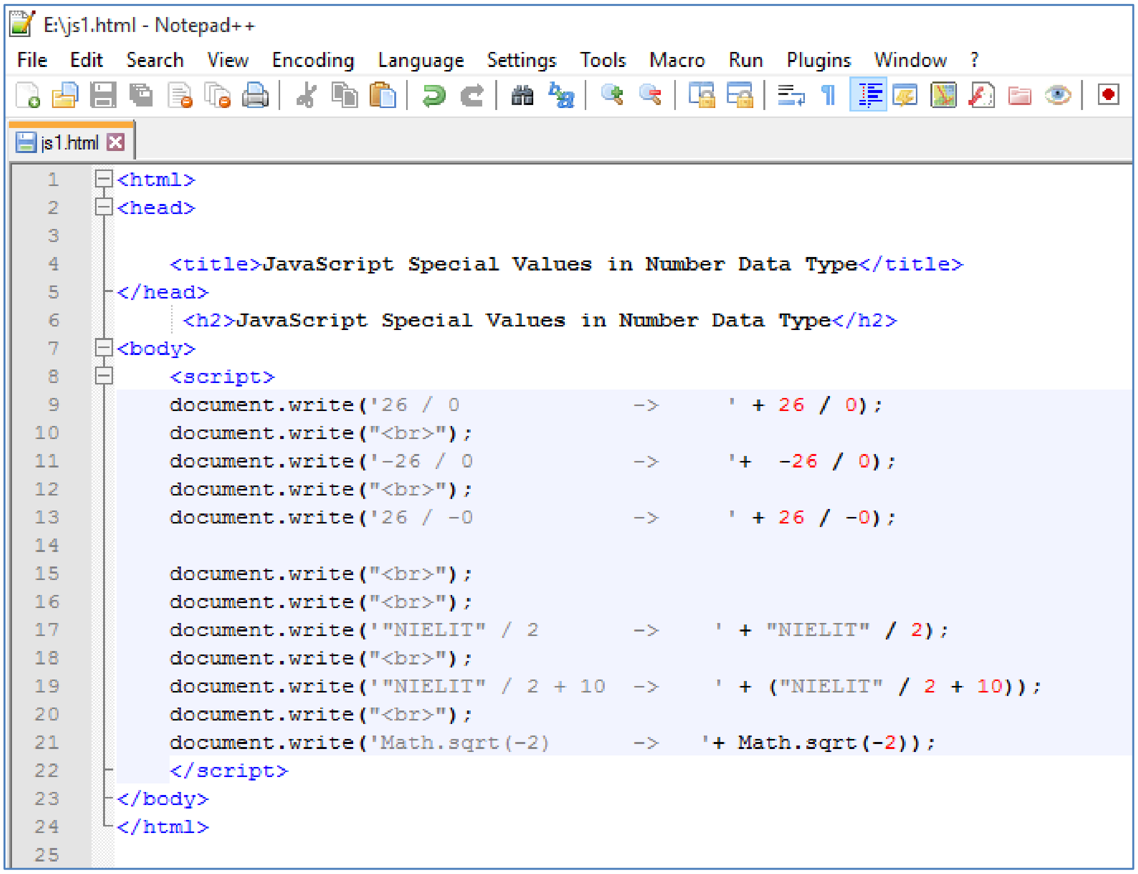
```
var a = 15;         // integer
var b = 25.5;       // floating-point number
var c = 4.27e+6;    // exponential notation, same as 4.27e6 or 4270000
var d = 4.27e-6;    // exponential notation, same as 0.00000427
```

- **Infinity, -Infinity** and **NaN** are the special values used with Number data type. Where
- **Infinity** represents ∞ which is greater than any number. Infinity is the result of dividing a nonzero number by 0
- **NaN** represents a special *Not-a-Number* value which is a result of an invalid or an undefined mathematical operation.

Example

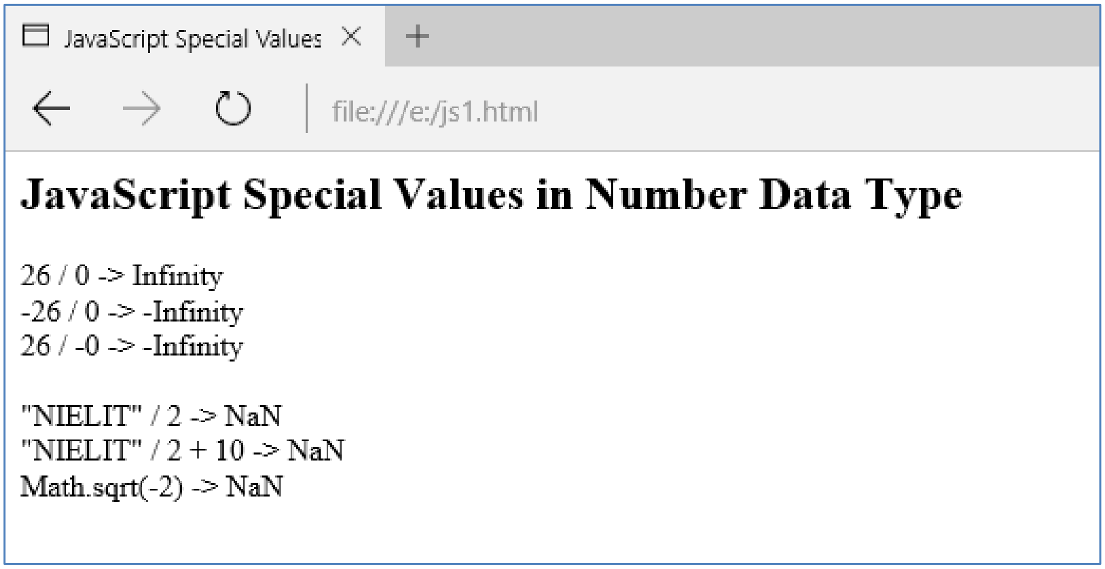
```
26 / 0;  // Output: Infinity
-26 / 0; // Output: -Infinity
26 / -0; // Output: -Infinity


“NIELIT" / 2;       // Output: NaN
"NIELIT" / 2 + 10;  // Output: NaN
Math.sqrt(-2);        // Output: NaN
```

Lets write these as in JavaScript code and execute:


E:\js1.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
js1.html

```html
<html>
<head>

    <title>JavaScript Special Values in Number Data Type</title>
</head>
     <h2>JavaScript Special Values in Number Data Type</h2>
<body>
    <script>
    document.write('26 / 0             ->      ' + 26 / 0);
    document.write("<br>");
    document.write('-26 / 0            ->      '+  -26 / 0);
    document.write("<br>");
    document.write('26 / -0            ->      ' + 26 / -0);

    document.write("<br>");
    document.write("<br>");
    document.write('"NIELIT" / 2       ->     ' + "NIELIT" / 2);
    document.write("<br>");
    document.write('"NIELIT" / 2 + 10  ->     ' + ("NIELIT" / 2 + 10));
    document.write("<br>");
    document.write('Math.sqrt(-2)      ->    '+ Math.sqrt(-2));
    </script>
</body>
</html>

```

Output



JavaScript Special Values in Number Data Type

```
26 / 0 -> Infinity
-26 / 0 -> -Infinity
26 / -0 -> -Infinity

"NIELIT" / 2 -> NaN
"NIELIT" / 2 + 10 -> NaN
Math.sqrt(-2) -> NaN
```

3. Boolean Data Type

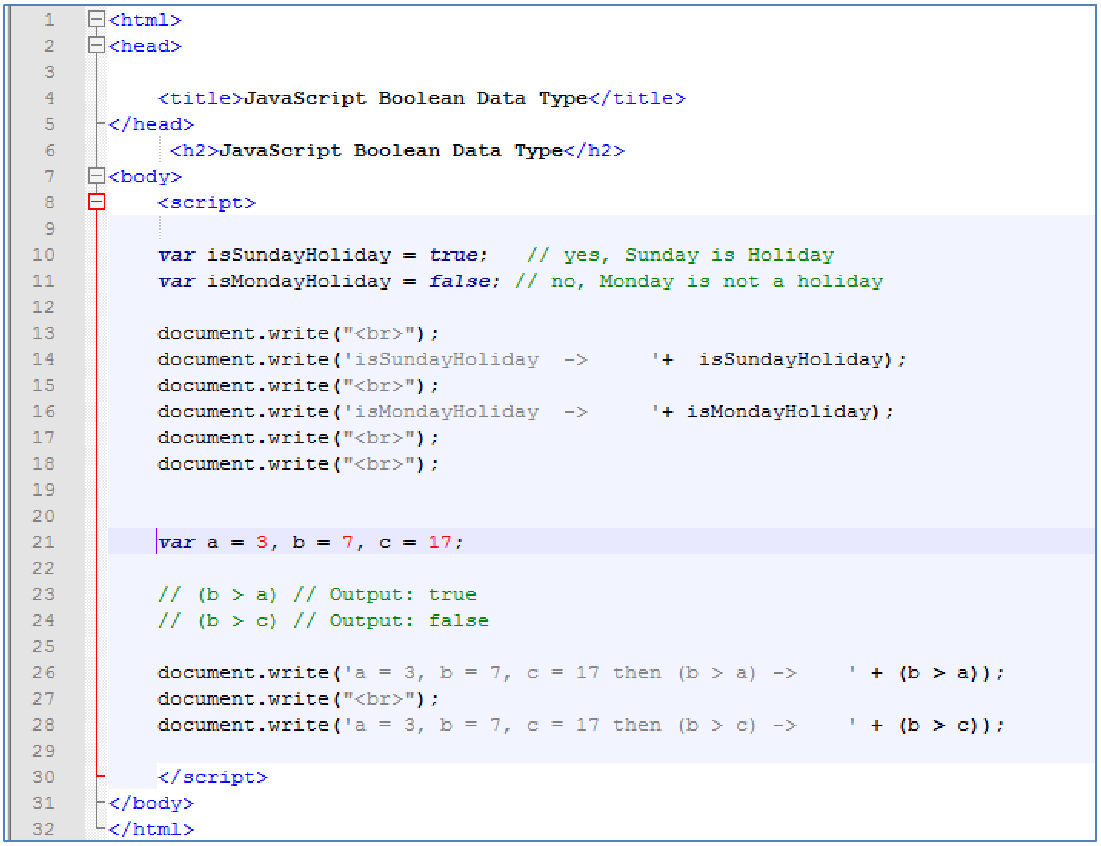
- The Boolean data type has only 2 output values : **true** or **false**.
- Boolean data types stores values like **Yes** (**true**) or **No** (**false**), **On** (**true**) or **Off** (**false**), etc.

- Boolean values may come as a result of comparisons in a program.

Example

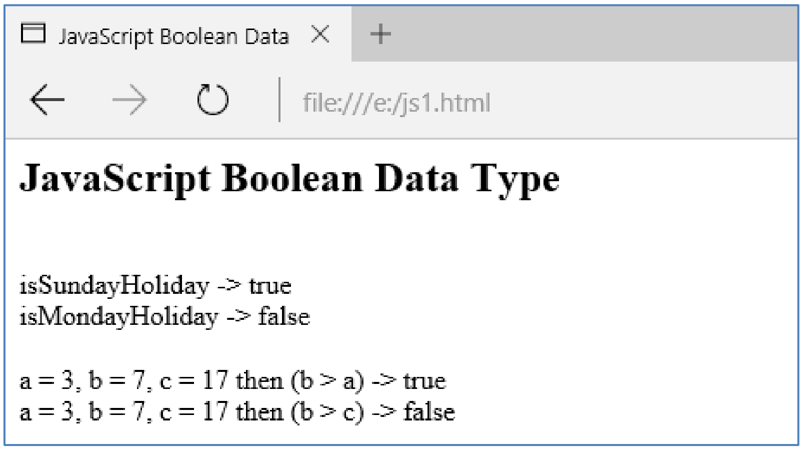
var isSundayHoliday = true; // yes, Sunday is Holiday
var isMondayHoliday = false; // no, Monday is not a holiday
var a = 3, b = 7, c = 17;

(b > a) // Output: true
(b > c) // Output: false



```
<html>
<head>

    <title>JavaScript Boolean Data Type</title>
</head>
     <h2>JavaScript Boolean Data Type</h2>
<body>
    <script>

    var isSundayHoliday = true;   // yes, Sunday is Holiday
    var isMondayHoliday = false; // no, Monday is not a holiday

    document.write("<br>");
    document.write('isSundayHoliday  ->     '+  isSundayHoliday);
    document.write("<br>");
    document.write('isMondayHoliday  ->     '+ isMondayHoliday);
    document.write("<br>");
    document.write("<br>");


    var a = 3, b = 7, c = 17;

    // (b > a) // Output: true
    // (b > c) // Output: false

    document.write('a = 3, b = 7, c = 17 then (b > a) ->    ' + (b > a));
    document.write("<br>");
    document.write('a = 3, b = 7, c = 17 then (b > c) ->    ' + (b > c));

    </script>
</body>
</html>
```

Output



JavaScript Boolean Data Type

```
isSundayHoliday -> true
isMondayHoliday -> false

a = 3, b = 7, c = 17 then (b > a) -> true
a = 3, b = 7, c = 17 then (b > c) -> false
```

Assignment

1. What are the different data types in JavaScript?
2. Explain the special values in number data types.