Expressions and Operators

- Operator is an object in computer programming that return a value by evaluating any given expression.
- Operators in JavaScript are classified into following groups

Assignment Operators:

- Assigns a value to its left operand based on the value of its right operand

Operator	Shorthand method	Meaning
Assignment	X = Y	X = Y
Addition Assignment	X += Y	X = X + Y
Subtraction Assignment	X -= Y	X = X – Y
Multiplication Assignment	X *= Y	X = X * Y
Division Assignment	X /= Y	X = X / Y

Remainder Assignment	X %= Y	X = X % Y
Exponent Assignment	X **= Y	X = X ** Y

Arithmetic Operators:

- Returns a number by handling any expression that evaluates a value.
- Arithmetic expression always returns a number.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus Division
**	Exponent [New in ES5] Early version use “Math.pow(number,power)” Ex: Math.pow(2,3); 2**3;

++	Increment x = x + 1
--	Decrement x = x - 1

Ex:

Pre and Post Increment

Post: It will assign and then increment or decrement.
X++

Pre: It will increment or decrement and then assign.
++X

Ex:

```
<script>
   function f1(){
      var x = 10;
      var y = x++;
      document.write(`X=${x}<br>Y=${y}`);        // x = 11, y=10
   }
   f1();
</script>
```

Ex:

```
<script>
   function f1(){
      var x = 10;
      var y = ++x;
      document.write(`X=${x}<br>Y=${y}`);      // x =
11, y=11
   }
   f1();
</script>
```

Ex:

```
<script>
   function f1(){
      var x = 10;
      var y = x--;
      document.write(`X=${x}<br>Y=${y}`);  //  x = 9,
y=10
   }
   f1();
</script>
```

FAQ:

String + String = string;

String + Number = String;

Number + Number = Number;

Number + Boolean = Number;

Boolean + Boolean = Number;

FAQ: Chaining

```
<script>
   function f1(){
      var x = 10;
      var y = 20;
      var z = (x = y);  // z=x=y;
      document.write(`Z=${z}`);     // Z = 20
   }
   f1();
</script>
```

Ex:

```
<script>
   function f1(){
```

```
        var x = 10;
        var y = 20;
        var z = 30;
        var a = (z=(x = y));
        document.write(`a=${a}`);
    }
    f1();
</script>
```

Ex:

```
<script>
    function f1(){
        var x = 10;
        var y = 20;
        var z=  30;
        var a = z += x *= y
        document.write(`a=${a}`);
    }
    f1();
</script>
```

Comparison Operators

Operator	Description
==	Equal [It can convert and compare]
!=	Not Equal
===	Strict Equal [Compare only same type values]
!==	String not equal.
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than equal

Ex:

```
<script>
   function f1(){
     var x = "10";
     var y = 10;
     document.write("x===y=" + (x===y));   // false
   }
   f1();
```

```
</script>
```

Ex:

```
<script>
    function f1(){
      var x = "10";
      var y = 10;
      document.write("x==y=" + (x==y));  // true
    }
    f1();
</script>
```

Logical Operators

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Special Operators

Operator	Description

typeof	It is used to verify and return the data type of value stored in a reference. Ex: <script> function f1(){ var product = { Name: "TV", Price: 45000.44, InStock: true, } document.write(` Type of Name : ${typeof product.Name}   Type of Price: ${typeof product.Price}   Type of Stock: ${typeof product.InStock} `); } f1();

	</script>
instance of	It is used to verify whether the given object derived from specific class and return true or false. EX: <script> class Employee { } function f1(){ var pic = new Image(); var products = new Array(); var emp = new Employee(); document.write(`Pic is Employee : ${pic instanceof Employee} `); document.write(`Pic is Image: ${pic instanceof Image} `); document.write(`Emp is Array: ${emp instanceof Array} `);

	document.write(`Emp is Employee: ${emp instanceof Employee} `); document.write(`Emp is Object: ${emp instanceof Object}`); } f1(); </script>

delete	It is used to delete any property from an object.
	You can’t delete properties of built in object.
	
	Ex:
	<script>
	class Employee {
	
	}
	function f1(){
	var product = {
	Name: "TV",
	Price: 45000.55

	}; var pic = new Image(); delete pic.src; // invalid delete Math.PI; // invalid delete product.Price; document.write(`Name=${product.Name }  Price=${product.Price} `); document.write(Math.PI); } f1(); </script>
in	It is used to check for a property in an object and return true if it is available. Syntax: “PropertyName” in ObjectName Ex: <script>

	function f1(){
	var product = {
	Name: "TV",
	Price: 45000.55
	};
	delete product.Price;
	document.write(`Is Price Available in Product :${"Price" in product} `)
	}
	f1();
	</script>

of	It is used to access a value from collection using iterator.
	It requires an iterator to read values from collection.
	
	Syntax:
	for(var item of collection) {
	}

	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	for(var item of products) {
	document.write(item + "
	");
	}
	}
	f1();
	</script>

new	It is dynamic memory allocating operator.
	It allocates memory for an object and loads its members into memory.
	
	Syntax:
	var products = new Array();
	var now = new Date();
	var pic = new Image();

?:	It is ternary operator. Syntax: (condition)?statement_true: statement_false Similar to “if..else”