# Numbers

- The Number() function converts the object argument to a number that represents the object's value.
- If the value cannot be converted to a legal number, NaN is returned.

var val = new Number(number);

Property

- NaN →a special value representing Not-a-Number
- MAX_VALUE →The largest possible value a number in JavaScript can have 1.7976931348623157E+308
- MIN_VALUE→The smallest possible value a number in JavaScript can have 5E-324
- NEGATIVE_INFINITY→A value that is less than MIN_VALUE.
- POSITIVE_INFINITY →A value that is greater than MAX_VALUE

# Methods

- toExponential(x) →Converts a number into an exponential notation
- toFixed(x) →Formats a number with a specific number of digits to the right of the decimal.
- toPrecision(x) →Defines how many total digits (including digits to the left and right of the decimal) to display of a number
- toString()→ Converts a Number object to a string
- valueOf() →Returns the primitive value of a Number object

```
var test4= 10 , test5= 20;
document.write(test4 +test5); //1020
document.write(Number(test4)+Number(test5)); //30
document.write(Number("99.66") +Number("01.34") ); //101
```

The parseInt() function parses a string and returns an integer.

```
document.write(parseInt("10.33") ) ;  //10
document.write(parseFloat("10.33") ); //10.33
```

# Array Object

- The Array object is used to store multiple values in a single variable.
- var myCars=new Array(); // regular array (add an optional integer
  myCars[0]="Saab"; // argument to control array's size)
  myCars[1]="Volvo";
  myCars[2]="BMW";
- var myCars=["Saab","Volvo","BMW"]; //literal array
- var myCars=new Array("Saab","Volvo","BMW"); // condensed array

```
var a=new Array(10);
for(var i=0;i<10;i++)
{
        a[i]=i+1;
        document.write(a[i]);
}
```

- Property
  - length→array **length** property returns an unsigned, 32-bit integer that specifies the number of elements in an array.
- Method
- concat(array1,array2,..)→Returns a new array comprised of this array joined with other array(s) and/or value(s).
- indexOf() →Returns the first (least) index of an element or -1.
- join(separator) →Joins all elements of an array into a string.
- lastIndexOf() →Returns the last (greatest) index of an element or -1
- pop() →Removes the last element from an array and returns that element.
- push(element1,…)→Adds one or more elements to the end of an array and returns the new length of the array.

- shift()→Removes the first element from an array and returns that element
- unshift(element1,…)→Adds one or more elements to the front of an array and returns the new length of the array.
- slice(bindx,eindx)→ Extracts a section of an array and returns a new array.
- splice(index, howMany, [element1],…)→ Adds and/or removes elements from an array
- sort()→ Sorts the elements of an array.
- toString()→ Returns a string representing the array and its elements.
- reverse()→ Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.
-

**Example 1:**

```
var parents = ["Jani", "Tove"];
var children = ["Cecilie", "Lone"];
var family = parents.concat(children);
document.write(family);    // Jani,Tove,Cecilie,Lone
```

**Example 2:**

```
var brothers = ["Stale", "Kai Jim", "Borge"];
var family = parents.concat(brothers, children);
document.write(family); //Jani,Tove,Stale,Kai Jim,Borge,Cecilie,Lone
```

**Example 3:**

```
var fruits = ["Banana", "Orange", "Apple" ];
document.write(fruits.join() + "<br >"); //Banana,Orange,Apple
document.write(fruits.join("+") + "<br >"); //Banana+Orange+Apple
document.write(fruits.join(" and ")); //Banana and Orange and Apple
```

**Example 4:**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.pop() ); //Mango //remove the last item
document.write(fruits );//Banana,Orange,Apple
document.write(fruits.push("Lemon","Pineapple") ); //5
document.write(fruits);//Banana,Orange,Apple, Lemon,Pineapple
document.write(fruits.reverse());//Pineapple, Lemon, Apple, Orange,
   Banana
document.write(fruits.shift() ); // Pineapple
document.write(fruits);// Lemon, Apple, Orange, Banana
document.write(fruits.unshift("Kiwi","Pineapple") ); //6
// added 5th and 6th item
document.write(fruits); // Kiwi,Pineapple,Lemon,Apple,Orange,Banana
```

**Example 5:**

```
Var fruits= [“Lemon”, “Apple”, “Orange”, “Banana”];
//Display from index 0 to index 2 (0 and1)
document.write(fruits.slice(0,2)); // Lemon, Apple
document.write(fruits.slice(1));//Apple, Orange, Banana //From 1st
document.write(fruits.slice(-2));// Orange, Banana //Last 2 items
```

**Example 6:**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.sort()); // Apple,Banana,Mango,Orange
var n = ["10", "5", "40", "25", "100", "1"];
document.write(n.sort()); //1,10,100,25,40,5
```

**Example 7:**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.toString()); //Banana,Orange,Apple,Mango
```

- **Deleting Elements**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
delete fruits[0];
```

- **Search an array for the item "Apple":**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var a = fruits.indexOf("Apple");
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var a = fruits.lastIndexOf("Apple");
```

- **Add items to the array at a positions:**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.splice(2, 0, "Lemon", "Kiwi");
```

- **At position 2, add the new items, and remove 1 item:**

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.splice(2, 1, "Lemon", "Kiwi");
```

- **Numeric Sort**

**Ascending**

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return a - b});
```

**Decending**

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b - a});
```

# For...In Statement

- The code in the body of the for…in loop is executed once for each property.

```
for (variable in object)
  {
  code to be executed
  }
var person={fname:"John",lname:"Doe",age:25};  //object creation
var x;
for (x in person)
{
document.write(person[x] + " "); // John Doe 25
}
```