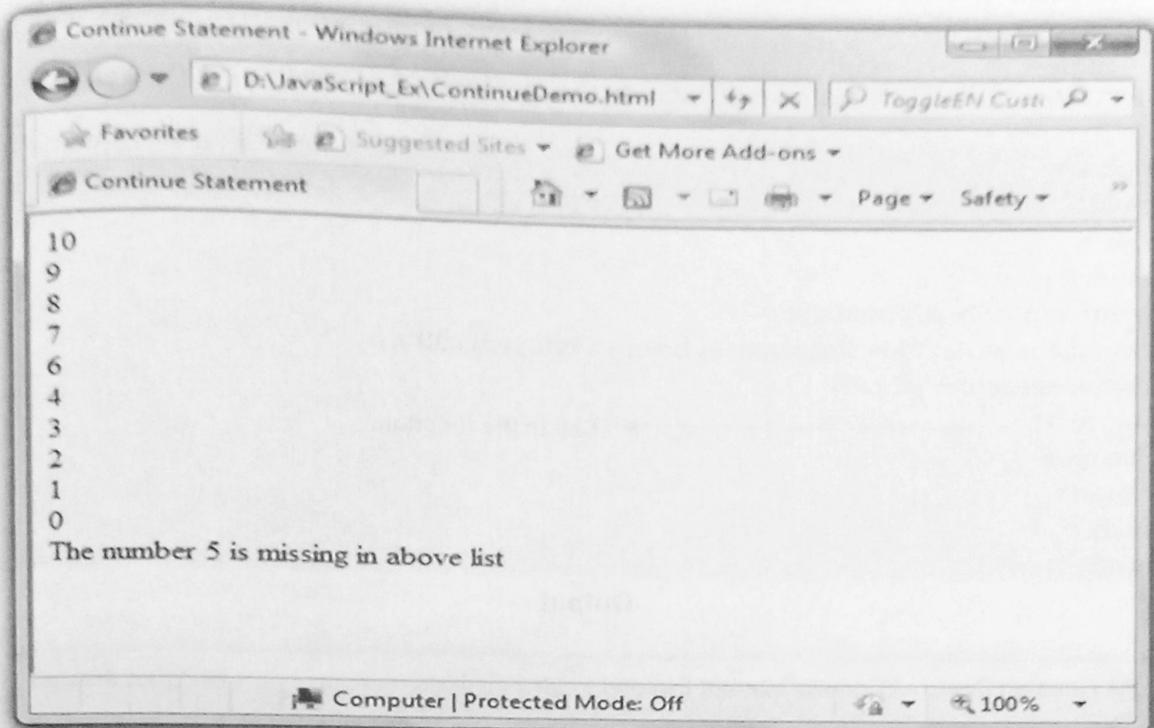


Output



4.9 Functions

We can write the functions in the JavaScript for bringing the modularity in the script. Separate functions can be created for each separate task. This ultimately helps in finding the bug from the program efficiently. We can define the function anywhere in the script either in head or body section or in both. But it is a standard practice to define the function in the **head** section and call that function from the **body** section. The keyword **function** is used while defining the function. The syntax for defining the function is

```
function name_of_function (arg1,arg2,...argn)
{
...
Statements
}
```

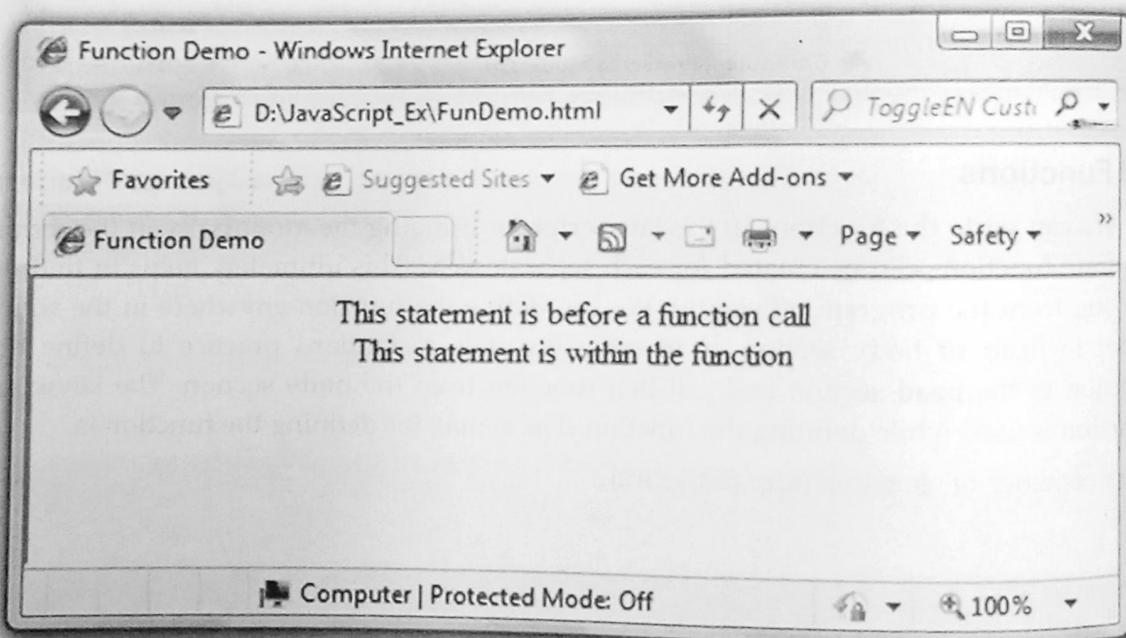
Here is a simple illustration in which we have written a function **my_fun()**

JavaScript[FunDemo.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Function Demo</title>
```

```
<script type="text/javascript">
  function my_fun()
  {
    document.write("This statement is within the function");
  }
</script>
</head>
<body>
<center>
<script type="text/javascript">
  document.write("This statement is before a function call");
  document.write("<br>");
  my_fun(); → Call to the function
</script>
</center>
</body>
</html>
```

Output

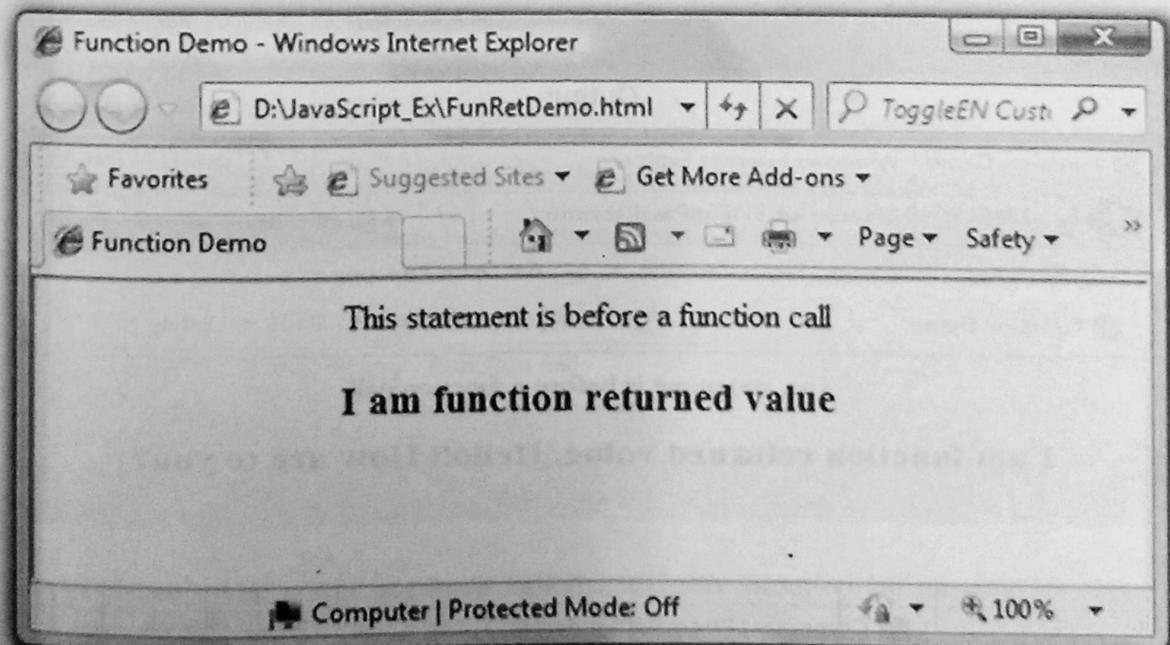


4.9.1 Returning Value from the Function

We can return some value from the function using a keyword **return**. The above script is slightly modified in which function returns something!

JavaScript[FunRetDemo.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Function Demo</title>
<script type="text/javascript">
function my_fun()
{
    str="I am function returned value";
    return str;
}
</script>
</head>
<body>
<center>
<script type="text/javascript">
document.write("This statement is before a function call");
document.write("<br>");
document.write("<h3>" + my_fun() + "<h3>");
</script>
</center>
</body>
</html>
```

Output

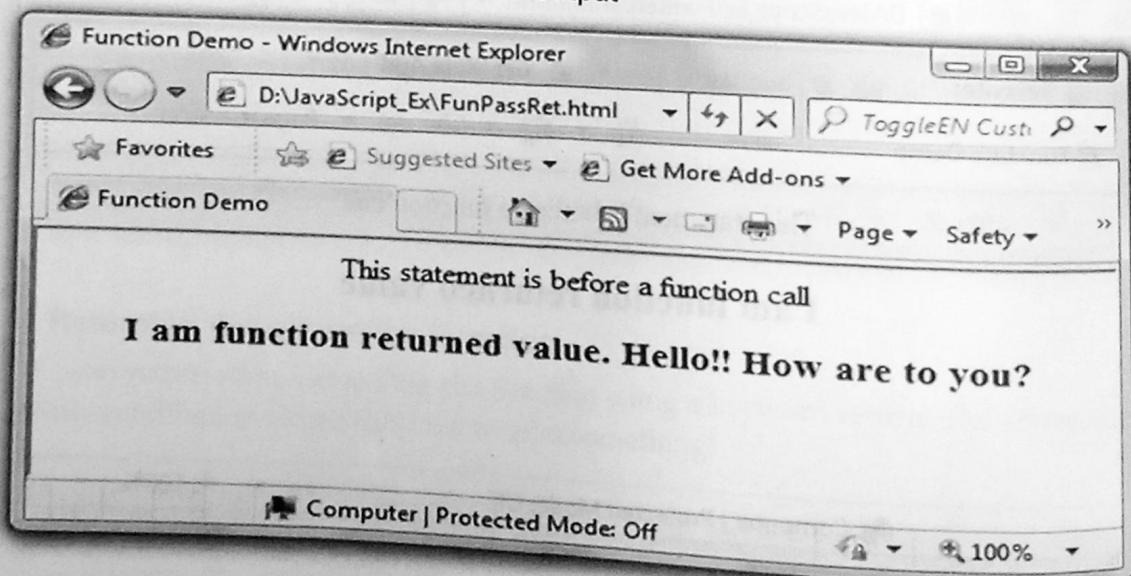
4.9.2 Passing the Parameters to the Function

Similarly we can pass some arguments to the function. In the following program, we have passed two arguments to the function and returning some value from the function.

JavaScript[FunPassRet.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>Function Demo</title>  
<script type="text/javascript">  
function my_fun(str1,str2)  
{  
    str="I am function returned value."+" "+str1+" "+str2 ;  
    return str;  
}  
</script>  
</head>  
<body>  
<center>  
<script type="text/javascript">  
document.write("This statement is before a function call");  
document.write("<br>");  
document.write("<h3>"+my_fun("Hello!!","How are to you?")+"<h3>");  
</script>  
</center>  
</body>  
</html>
```

Output



4.9.3 Passing an Array to the Function

Similar to C or C++ we can pass an entire array as a parameter to the function. This method of array passing is called as **call by reference**. When an array is passed to the function the array name is simply passed. In the following JavaScript we have initialized an array in the body part and to display those contents of an array we have called a `display()` function. And to this function an array is passed as an argument. Let see the JavaScript.

JavaScript[FunPassArr.html]

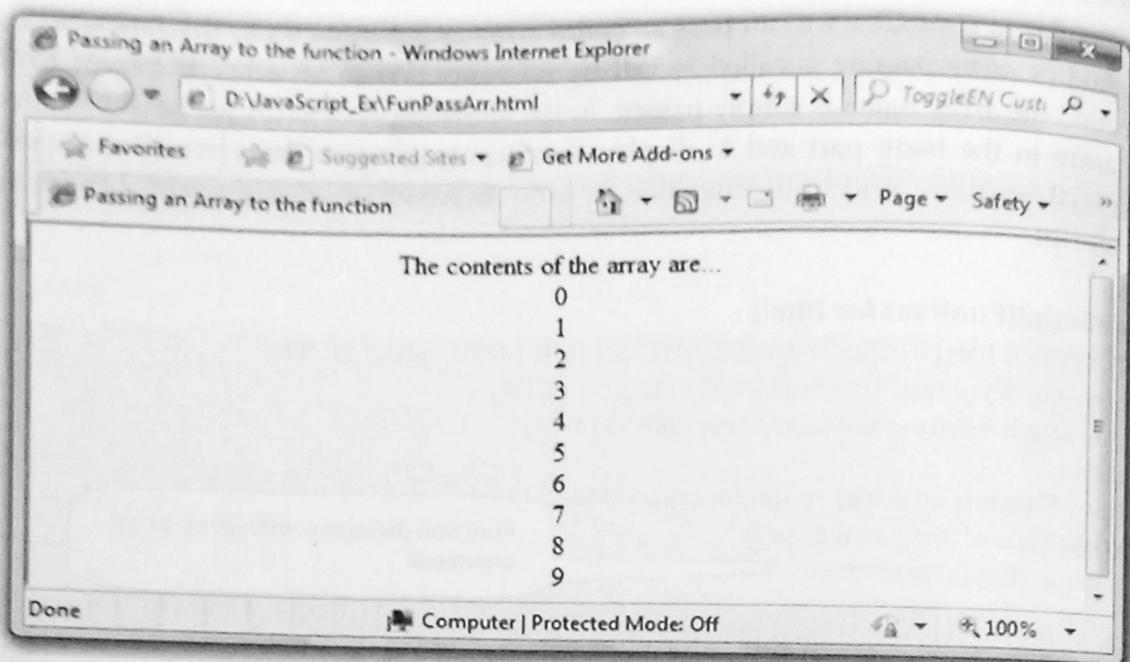
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Passing an Array to the function</title>
<script type="text/javascript">
function display(a)
{
    document.write("The contents of the array are..." + "<br>");
    i=0;
    for(i in a)
    {
        document.write(a[i] + "<br>");
        i++;
    }
}
</script>
</head>
<body>
<center>
<script type="text/javascript">
var ar=new Array(10);
for(i=0;i<=9;i++)
{
    ar[i]=i;
}
display(ar);
</script>
</center>
</body>
</html>
```

Function definition with array as an argument

Function call

In above given script we have called a `display` function by passing an array as a parameter to it. Hence we will get following output.

Output



Let us see one more example in which we are sorting the elements of an array using function.

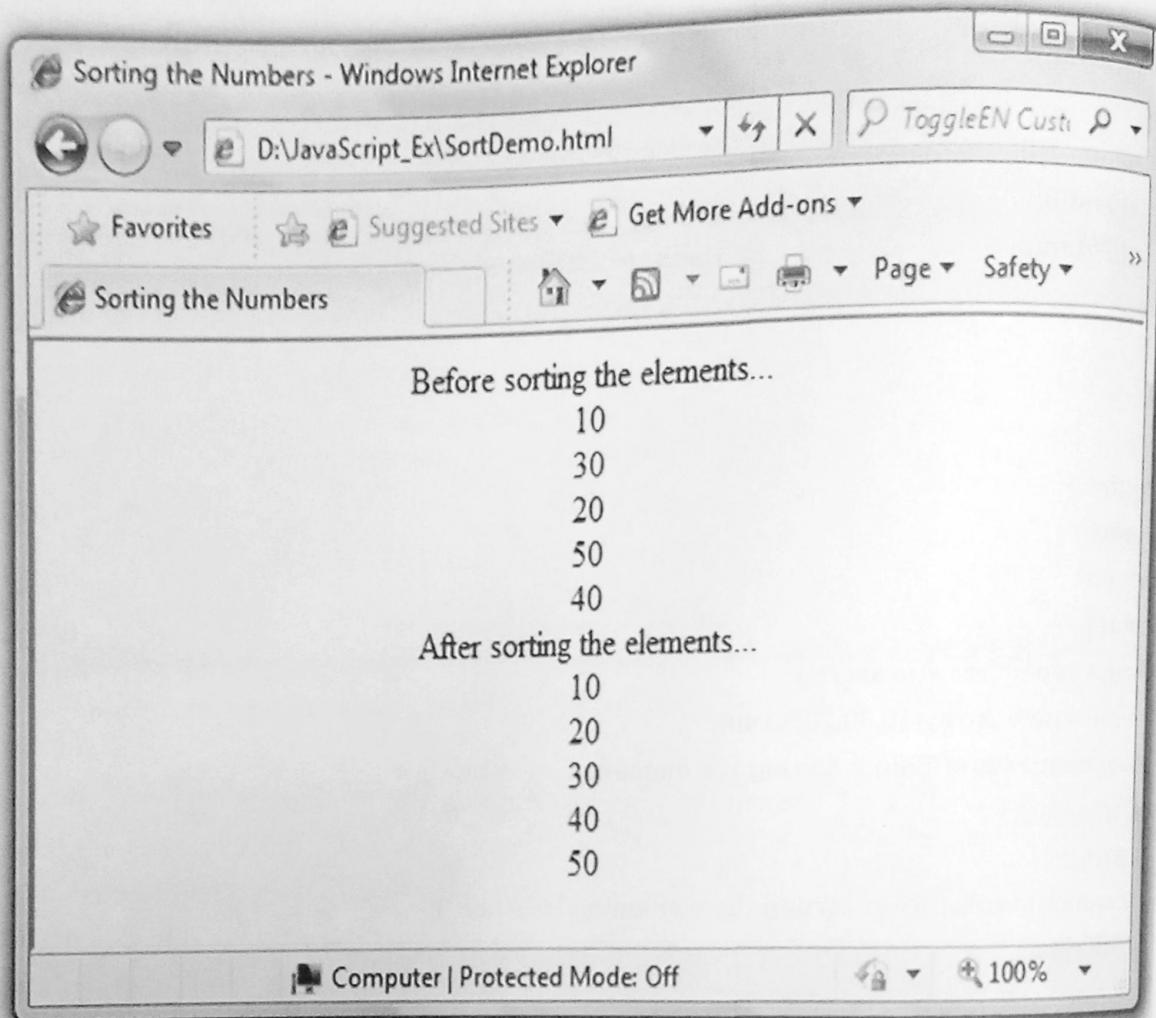
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Sorting the Numbers</title>
<script type="text/javascript">
function display(a)
{
    i=0;
    for(i in a)
    {
        document.write(a[i]+"<br>");
        i++;
    }
}
function sort(a)
{
    for(i=0;i<a.length-1;i++)
    {
```

```
for(j=i+1;j<a.length;j++)
{
    if(a[j]>a[i])
    {
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
    }
}
}
}

</script>
</head>
<body>
<center>
<script type="text/javascript">
var ar=new Array(10,30,20,50,40);
document.write("Before sorting the elements..."+<br>);
display(ar);
sort(ar);
document.write("After sorting the elements..."+<br>);
display(ar);
</script>
</center>
</body>
</html>
```

In the above program we have stored some values in the array in the body part then in order to display the contents of an array we have called **display** function then we have called one more function called **sort** in which the elements of an array are sorted. Note that in the **sort** function we have used *length* property which returns total number of elements in an array. The simple bubble sort method is used to sort the array in an ascending order. Again the **display** function is called in order to display the sorted array.

Output



4.10 Arrays

Arrays is a collection of similar type of elements which can be referred by a common name. Any element in an array is referred by an **array name** followed by "[" followed by **position** of the element followed by]. The particular position of element in an array is called **array index or subscript**.

For example -

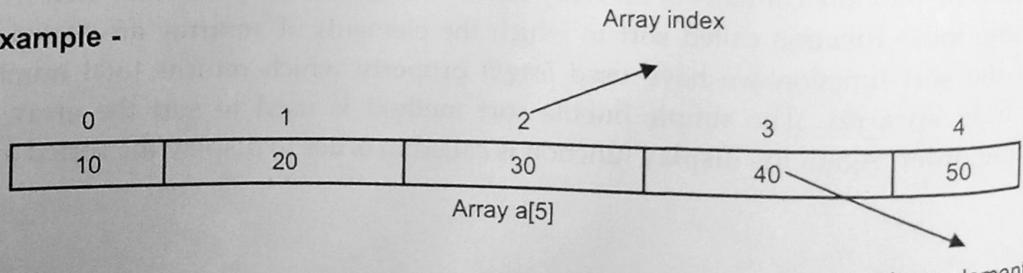


Fig. 4.2 Arrays

Normally the first element in an array is stored at 0th location, however we can start storing the element from any position.

4.10.1 Array Declaration

In JavaScript the array can be created using **Array** object. We will get introduced with the concept of objects in the later part of this chapter. Let us now understand how to declare and create an array. Suppose, we want to create an array of 10 elements then we can write,

```
var ar = new Array(10);
```

Using **new** operator we can allocate the memory dynamically for the arrays. In the brackets the size of an array is mentioned and the **var ar** denotes the name of the array. Thus by the above sentence an array ar will be created in which we can store 10 elements at the most. Sometimes the above statement can be written like this

```
var ar;  
ar=new Array(10);
```

4.10.2 Array Initialization

Let us see how to store some elements in an array.

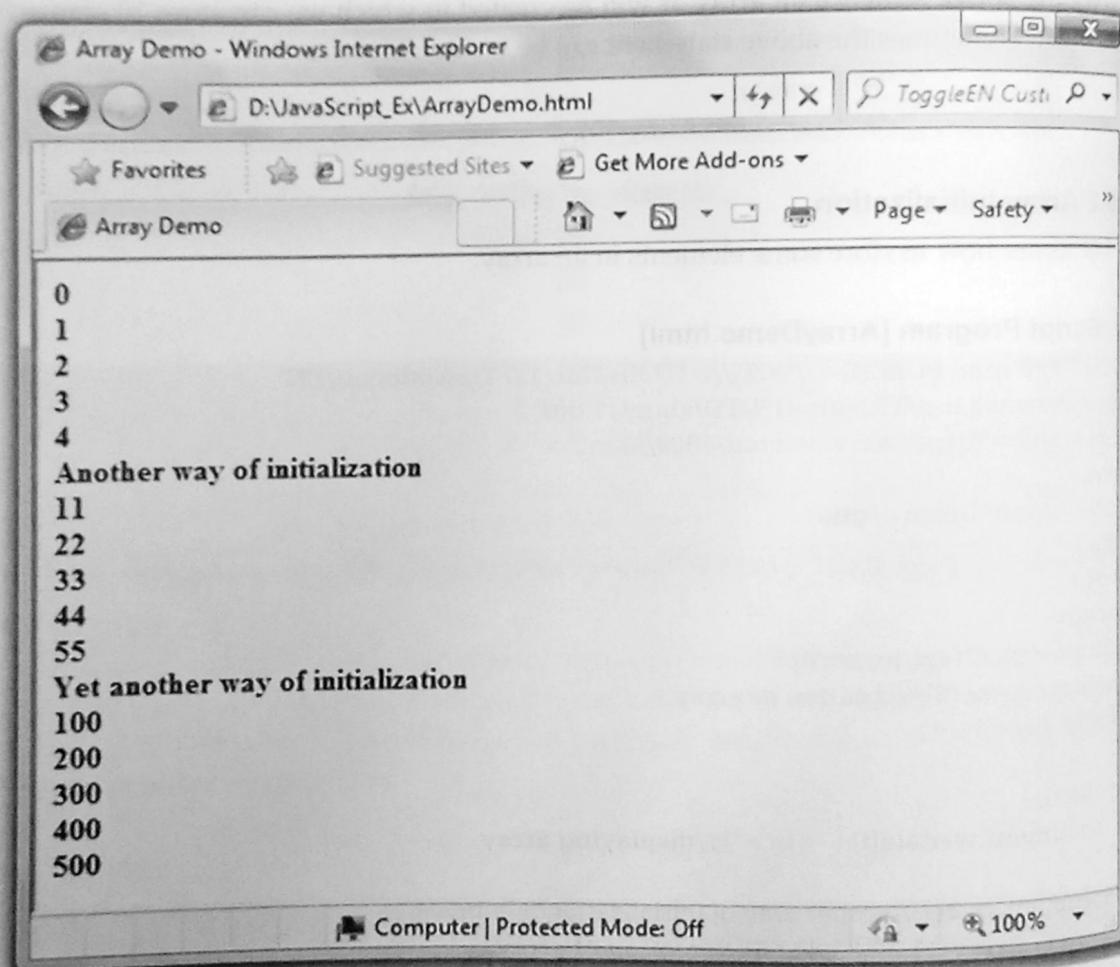
JavaScript Program [ArrayDemo.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <title>Array Demo</title>  
</head>  
<body>  
<strong>  
  <script type="text/javascript">  
    a=new Array(5);//creation of array  
    for(i=0;i<5;i++)  
    {  
      a[i]=i;  
      document.write(a[i]+<br>());//displaying array  
    }  
    document.write("Another way of initialization"+<br>());  
    b=new Array(11,22,33,44,55);//creation of array  
    for(i=0;i<5;i++)  
    {  
      document.write(b[i]+<br>());//displaying array  
    }  
    document.write("Yet another way of initialization"+<br>());  
    var c=[100,200,300,400,500];//creation of array
```

```
for(i=0;i<5;i++)
{
    document.write(c[i]+"<br>");//displaying array
}
</script>
</strong>
</body>
</html>
```

As you can notice that, in above JavaScript an array can be initialized in three different ways which is shown by boldface. Hence an output of above script will be

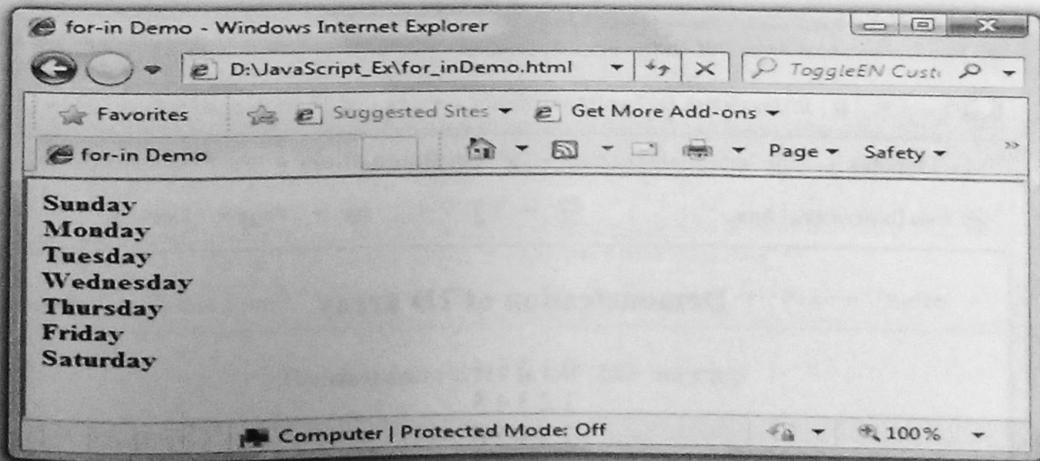
Output



There is one control structure in JavaScript which is closely associated with array elements and such a control structure is **for...in**. Let us see a simple JavaScript which makes use of **for-in** control structure to display elements of an array.

JavaScript [for_inDemo.html]

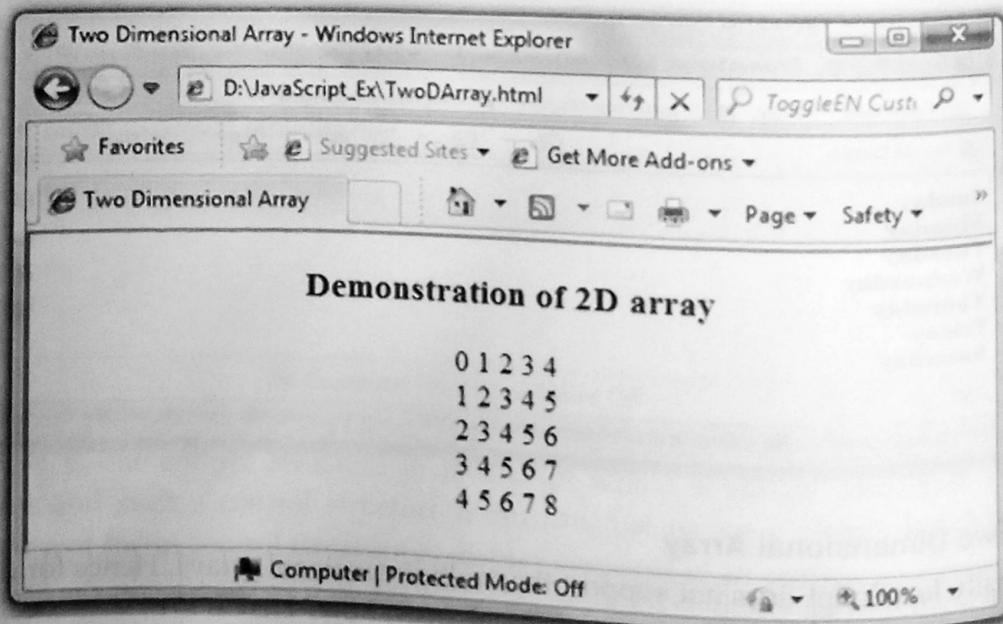
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>for-in Demo </title>
</head>
<body>
<strong>
<script type="text/javascript">
Days=new Array();
Days[0]="Sunday";
Days[1]="Monday";
Days[2]="Tuesday";
Days[3]="Wednesday";
Days[4]="Thursday";
Days[5]="Friday";
Days[6]="Saturday";
for(i in Days)
{
document.write(Days[i]+<br>);
}
</script>
</strong>
</body>
</html>
```

Output**4.10.3 Two Dimensional Array**

Actually JavaScript does not support the multidimensional arrays. Hence for defining the 2D array we make use of single dimensional array, how? Here it is -

JavaScript[TwoDArray.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Two Dimensional Array</title>
</head>
<body>
<center>
<h3> Demonstration of 2D array</h3>
<script type="text/javascript">
a=new Array();//creation of rows of array
for(i=0;i<5;i++)
a[i]=new Array();//creating columns of array
for(i=0;i<5;i++)
{
for(j=0;j<5;j++)
{
a[i][j]=i+j;
document.write(a[i][j] + " ");
}
document.write("<br>");
}
</script>
</center>
</body>
</html>
```

Output

There is another way by which the two dimensional array can be initialized. This method is represented in the following Script

JavaScript[TwoDArray1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Two Dimensional Array</title>
</head>
<body>
<center>
<h3> Demonstration of 2D array</h3>
<script type="text/javascript">
var a=[ [1,2,3],
        [4,5,6],
        [7,8,9]
    ]; //creation of array
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
document.write(a[i][j] + " ");
}
document.write("<br>");
}
</script>
</center>
</body>
</html>
```

Output

