

Client Side Scripting





CO2

- Implement Arrays and functions in Java script

LO1

- Create Array to solve the given problem



TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory						Practical						
				Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25#	10	25	10	50	20



What is an array?

How to create an array in Javascript?



CONTENTS

1. Declaring an array
2. Initializing an array
3. Defining an arrays elements
4. Looping an array
5. Adding an array element
6. Sorting an array element
7. Combining an array elements into a string
8. Changing elements of an array
9. Objects as associative arrays

DECLARING AN ARRAY



- An array is an object that can store a collection of items.
- Arrays are useful when you need to store large amount of data of the same type like storing of details of 100 students.
- You can access the items in an array by referring to its index number and the index of the first element of an array is 0
- An array in Javascript can hold different elements – we can store numbers, strings and Boolean in a single array.
- There are basically two ways to declare an array

1. By array literal

2. By creating instance of array

1. BY ARRAY LITERAL

- **Syntax:**

```
Var array name = [value1,value2.....valueN ];
```

- Values are contained inside[] and separated by, (comma).

Program: Write a Javascript code to demonstrate Array Literal.

```
<html>
<head>
<title>Array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a =[1,2,3,4,5];
document. Write("length of array =" +a.length);
</script>
<body>
</body>
</html>
```

Output: length of array =5



2. BY CREATING INSTANCE OF ARRAY



- **Syntax:**

```
Var array name = new Array();
```

- New keyword is used to create instance of array.

Program: Write a Javascript code to create an Array and print its length.

```
<html>
<head>
<title>Array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new Array(3);
document. Write("length of array =" +a.length);
</script>
<body>
</body>
</html>
```

Output: length of array =3

INITIALIZING AN ARRAY



- Initialization is the process of assigning values to an array.
- While initializing an array , all the elements should be placed in parenthesis and separated by commas.

Program: Write a Javascript code to create and initialize an element in array.

```
<html>
<head>
<title>Array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new Array("c","c++","java","vb");
document. Write("length of array =" +a.length);
</script>
<body>
</body>
</html>
```

Output: length of array =4



DEFINING AN ARRAY ELEMENTS

- Array contains a list of elements, each element in the array is identified by its index.
- The first element in an array stores at 0th position, second element at 1st position, third element at 2nd position and so on.
- Assignment (=) operator is used to assign values to an array elements.
- Elements can be retrieving by its index position.

DEFINING AN ARRAY ELEMENTS - EXAMPLE



Program: Write a Javascript code to create and print an element in Array.

```
<html>
<head>
<title>Array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new Array(3);
a[0]="c";
a[1]="c++";
a[2]="java";
a[3]="vb";
document. Write("<br>elements at 0th place="+a[0]);
document. Write("<br>elements at 1st place="+a[1]);
document. Write("<br>elements at 2nd place="+a[2]);
document. Write("<br>elements at 3rd place="+a[3]);
</script>
<body>
</body>
</html>
```

LOOPING AN ARRAY



- In Javascript, **for loop** iterate over each item in an array.
- Array are zero based, which means the first item is referenced with an index of 0.
- Elements in array are access by a numeric index, starting at zero and ending with the array length minus 1.

LOOPING AN ARRAY - EXAMPLE



Program: Write a Javascript code to create and print an element in array using for loop.

```
<html>
<head>
<title>Array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new Array(4);
a[0]="c";
a[1]="c++";
a[2]="java";
a[3]="vb";
For(var i=0; i<a.length; i++)
{
document. Write("<br>element="+a[i]);
}
</script>
<body>
</body>
</html>
```



ADDING AN ARRAY ELEMENT

- It is important to increase the size of an array by adding elements.
- There are two ways to add elements in an array:

1. Using push () method

2. Using unshift () method

1. Using push() method :

- It adds one or more elements at the end of an array.
- This method changes the original length of an array.

- **Syntax:**

```
array. Push(element1,element2....element-n)
```

1. USING PUSH() METHOD - EXAMPLE



Program: Write a Javascript code to implement push method of array.

```
a.push("C");
a.push("C++","JAVA");
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new array();
a[0]="C";
a[1]="C++";
document. Write("original length="+a.length + "<br>");
a.push("JAVA");
a.push("VB", "PHP");
document. Write("new length after push method="+a.length + "<br>");
For(var i=0; i<a.length ; i++)
{
document. Write(a[i]);
}
</script>
<body> </body> </html>
```



2. USING UNSHIFT() METHOD

- It adds one or more elements in the beginning of an array.
- It returns the updated array with change in length.

- **Syntax:**

```
array.unshift(element1,element2....element-n)
```


2. USING UNSHIFT() METHOD - EXAMPLE



Program: Write a Javascript code to add elements in array using unshift method of array.

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a = new array();
a[0]="JAVA";
document. Write("original length="+a.length +"<br>");
a.unshift("C","C++","VB");
document. Write("new length after unshift method="+a.length +"<br>");
For(var i=0; i<a.length ; i++)
{
document. Write(a[i]);
}
</script>
<body>
</body>
</html>
```

SORTING AN ARRAY ELEMENT



There are two functions are used for sorting an array element:

1. **sort()**
2. **reverse()**

1. **sort() :**

- sorts the elements of an array in place and returns the sorted array.
- When `sort()` without arguments is called, it will sort elements in alphabetical order.

- **Syntax:**

```
array. Sort (compareFunction);
```

- The `sort()` function is used to sort the array in place in a given order according to the `compare()` function.
- The only argument to this function is a compare function that is used to sort the elements according to different attributes and in the different order.
 - `compareFunction(a,b) < 0` = then a comes before b in the answer.
 - `compareFunction(a,b) > 0` = then b comes before a in the answer.
 - `compareFunction(a,b) = 0` = then the order of a and b remains unchanged.

SORTING AN ARRAY ELEMENT – SORT() FUNCTION



Program : Write a Javascript code to sort elements in array:

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]=4;
a[1]=3;
a[2]=1;
a[3]=2;
a.sort();
For(var i=0; i<a.length; i++)
{
document. Write(a[i]);
}
</script>
<body>
</body>
<html>
```

Output: 1234

SORT() FUNCTION - EXAMPLE

Program : Write a Javascript code to sort array elements in ascending order:



```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]=20;
a[1]=13;
a[2]=5;
a[3]=2;
a.sort(function(a,b) {return a-b});
For(var i=0; i<a.length; i++)
{
document. Write(a[i] + "<br>");
}
</script>
<body>
</body>
<html>
```

Output: 2 5 13 20

SORTING AN ARRAY ELEMENT – REVERSE() FUNCTION



2. Reverse():

- Sort method will sort all elements from array in ascending order.
- we can use reverse method to display sorted list in reverse manner i.e. in descending order.

- **Syntax:**

```
array.reverse( );
```

- It will return the reversed single value of the array.

REVERSE() FUNCTION - EXAMPLE



Program : Write a Javascript code to sort array elements in descending order:

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]=5;
a[1]=3;
a[2]=4;
a[3]=2;
a[4]=1;
Var r=a.sort();
r.reverse();
For(var i=0; i<a.length; i++)
{
document. Write(a[i] + "<br>");
}
</script>      <body>      </body>
<html>
```

Output : 5 4 3 2 1

COMBINING AN ARRAY ELEMENTS INTO STRINGS



In Javascript an array can be combined into string using two functions:

1. **Join()**
2. **Concat()**

1. Join() : This function joins all elements of an array into string.

- **Syntax:**

```
array.join(separator);
```

- Here separator is a string like / , - , * etc to separate each element of the array.
- If separator is not mentioned , it will display list with comma as separator.
- This function returns a string after joining all elements together.

JOIN() FUNCTION WITHOUT SEPARATOR - EXAMPLE



Program : Write a Javascript code to combine array elements into string without separator.

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]="C";
a[1]="C++";
a[2]="JAVA";
Var r=a.join();
document. Write("after joining=" +r);
</script>
<body>
</body>
</html>
```

Output : after joining=C,C++,JAVA



JOIN() FUNCTION USING SEPARATOR - EXAMPLE

Program : Write a Javascript code to combine array elements into string using separator.

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]="C";
a[1]="C++";
a[2]="JAVA";
Var r=a.join("/");
document. Write("after joining=" +r);
</script>
<body>
</body>
<html>
```

Output : after joining=C/C++/JAVA

2. CONCAT() FUNCTION



- This function is used to join two or more array together in Javascript.
- this function returns a new string which is combination of different strings passed to it as arguments.

- **Syntax:**

```
string1.concat(string2, string3,.....string N );
```

2. CONCAT() FUNCTION - EXAMPLE



Program : Write a Javascript code to implement Concat method of array.

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]="C";
a[1]="C++";
a[2]="JAVA";
Var r=a.concat("VB");
document. Write("after joining=" +r);
Var r1=a.concat("NET","PHP");
document. Write("<br>after joining=" +r1);
</script>
<body>
</body>
</html>
```

**Output : after joining=C,C++,JAVA,VB
after joining=C,C++,JAVA,NET,PHP**

CHANGING ELEMENTS OF AN ARRAY



- In Javascript , following methods are used for changing elements of an array

1. Shift()

2. Pop()

3. Splice()

1. Shift(): This method is used to remove first element from an array and returns the removed single value of an array.

- **Syntax:**

```
array.shift( );
```

1. SHIFT() METHOD - EXAMPLE

Program : Write a Javascript code to implement shift method of array.



```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]=5;
a[1]=3;
a[2]=4;
a[3]=2;
a[4]=1;
document. Write("elements before shift...<br>");
For(var i=0; i<a.length;i++)
{
document. Write(a[i]+"<br>");
}
</script>
<body>
</body>
</html>
```



2. POP() METHOD:

- This method is used to remove last element and returns the removed element from an array.

- **Syntax:**

```
array.pop( );
```

2. POP() METHOD - EXAMPLE



Program : Write a Javascript code to implement pop method of array.

```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript" type="text/javascript">
Var a= new array();
a[0]=5;  a[1]=3;  a[2]=4;  a[3]=2;  a[4]=1;
document. Write("elements before pop...<br>");
For(var i=0; i<a.length;i++)
{
document. Write(a[i]+"<br>");
}
a.pop();
document. Write("elements after pop...<br>");
For(var i=0; i<a.length;i++)
{
document. Write(a[i]+"<br>");
}
</script>
<body>
</body>
</html>
```



3. SPLICE() METHOD:

- This method is used to add or remove the elements to or from the existing array.
- It returns the removed elements from an array.
- The splice() method also modifies the original array.

- **Syntax:**

```
array. Splice(start,delete,element1,element2,...element n );
```

- **Start** : It represents the index from where the method start to extract the elements.
- **Delete** : It is optional. It represents the number of elements to be removed.
- **Element1,element2,element3,...element n** : It is optional. It represent the elements to be inserted.

SPLICE() METHOD – EXAMPLE-1

Program: Write a Javascript code to add element in array using splice method.



```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript " type="text/javascript">
Var a=new array();
a[0]="C";
a[1]="C++";
document.write("original length=" + a.length+"<br>");
a.splice(2, 0, "JAVA", "PHP");
document.write("new length=" +a.length + "<br>");
For(var i=0; i<a.length ; i++)
{
document.write(a[i]);
}
</script>
<body>
</body>
</html>
```

SPLICE() METHOD – EXAMPLE-2

Program: Write a Javascript code to remove element from array using splice method.



```
<html>
<head>
<title>array demo</title>
</head>
<script language="Javascript " type="text/javascript">
Var a=new array();
a[0]="C";
a[1]="C++";
a[2]="JAVA";
a[3]="VB";
document.write("original length=" + a.length+"<br>");
a.splice(2);
document.write("new length=" +a.length + "<br>");
For(var i=0; i<a.length ; i++)
{
document.write(a[i]);
}
</script>
<body> </body> </html>
```



OBJECTS AS ASSOCIATIVE ARRAY

- Associative arrays are dynamic objects that the user redefines as needed.
- When you assign values to keys in a variable of type array, the array is transformed into an object, and it loses the attributes and methods of array.
- We can create it by assigning a literal to a variable.

- **Syntax:**

```
array={key1 : 'value1', key2 : 'value2'};
```

- **Example:**

```
var a ={"one" : 1, "two" : 2 ,"three": 3};
```

- The value is stored in association with key and the value is accessed by keys.

```
var r = a["one"];
```



OBJECTS AS ASSOCIATIVE ARRAY – EXAMPLE-1

Program: Write a Javascript code to demonstrate objects as associative array.

```
<html>
<head>
<title>Associative array demo</title>
</head>
<script language="Javascript " type="text/javascript">
Var a= {"name": "john","age": 23,"language": "english"};
document.write("name="+a["name"]);
document.write("<br>age="+a["age"]);
document.write("<br>language="+a["language"]);
</script>
<body>
</body>
</html>
```

**Output: name: john
age:23
language: English**

OBJECTS AS ASSOCIATIVE ARRAY – EXAMPLE-2



- We can create an associative array with the **Object** reserved word , then assign keys and values.

Program: Write a Javascript code to demonstrate objects as associative array.

```
<html>
<head>
<title>Associative array demo</title>
</head>
<script language="Javascript " type="text/javascript">
Var a=new Object();
a["name"]="john";
a["age"]=23;
a["language"]="english";
For (var i in a)
{
document.write(i+"="+a[i]+"<br>");
}
</script>
<body>
</body>
</html>
```

OBJECTS AS ASSOCIATIVE ARRAY USING PROPERTY



- **Syntax:**

```
array. Key=value;
```

Program: Write a Javascript code to demonstrate objects as associative array.

```
<html>
<head>
<title>Associative array demo</title>
</head>
<script language="Javascript " type="text/javascript">
Var a=new Object();
a.name="john";
a.age=23;
a.language="English";
For (var i in a)
{
document.write(i+"="+a[i]+"<br>");
}
</script>
<body>
</body>
```



QUIZ TIME

Q1. Which is the correct way to write a JavaScript array?

- A. `var txt = new Array(1:"tim",2:"kim",3:"jim")`
- B. `var txt = new Array:1=("tim")2=("kim")3=("jim")`
- C. `var txt = new Array("tim","kim","jim")`
- D. `var txt = new Array="tim","kim","jim"`

► **Ans. C.**

`var txt = new Array("tim","kim","jim")`

Q2 .

- (a) short
- (b) int
- (c) long
- (d) All the above

► **Ans. d. All the above**

QUIZ TIME



Q3. Choose a correct statement about C language arrays

- a) An array address is the address of first element of array itself.
- b) An array size must be declared if not initialized immediately.
- c) Array size is the sum of sizes of all elements of the array.
- d) All of the above

► **Ans. d. All of the above**

• Q4. An array Index starts with.?

- a) -1
- b) 0
- c) 1
- d) 2

► **Ans. b. -1**



Thank You