

2. Array, function & string



S.N.J.B.'s
SHRI H. H. J. B. POLYTECHNIC, CHANDWAD

CLASS TEST I/II (201 - 201)

Course & Semester :		Roll No.:	Date : / /201
Name of Subject :		Marks obtained :	
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

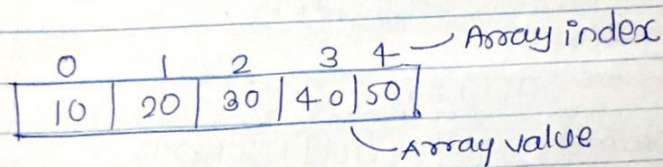
0592

* Array :-

- It is collection of similar type of elements which can be referred by common name.

- Any element in array is referred by array name followed by [followed by position of the element followed by].

- The particular position of element in array is called array index or subscript.



Declr :-

array can be created using Array object

for eg:-

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

using new operator we can allocate the memory dynamically for the arrays.

or

```
var a;
```

```
a = new Array(10);
```

Initializing

initialization is the process of assigning value when either variable or array is declared.

- 1) Declare name of array
- 2) Make use of new keyword.
- 3) Each value within array as array element must be separated by comma.

eg:-

```
var a = new Array(10, 20, 30, 40, 50);
```

elements in array are stored from index 0.

For eg:-

```
<html>
<body>
<script type="text/javascript">
```

```
    a = new Array(5);
    for (i=0; i<5; i++)
    {
        a[i] = i;
        document.write(a[i] + "<br>");
    }
```

```
    b = new Array(11, 12, 13, 14, 15);
    for (i=0; i<5; i++)
    {
        document.write(b[i]);
    }
```

```
    var len = b.length;
    document.write("length of array is " + len);
    var c = [100, 200, 300, 400, 500];
    for (i=0; i<5; i++)
    {
        document.write(c[i]);
    }
</script>
</body>
</html>
```

Looping an Array - Visiting each element present in array

```
<html>
<body>
<script type="text/javascript">
    Days = new Array();
    Days[0] = "sunday";
```

```
    Days[6] = "saturday";
    for (i=0; i<Days.length; i++)
    {
        document.write(Days[i]);
```

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

Sunday
Monday
Tuesday
Wednesday

Adding an Array element

above code same for (10, 20, 30, 40, 50, 60, 70)
& then next statement

```
a[a.length] = 80;
for (i=0; i<a.length; i++)
{
    document.write(a[i] + " ");
```

10 20 30 40 50
60 70 80

& then next close script tag..

Sorting array element :-

```
A.sort();
then print array
```

eg: 40 30 10 50 20
10 20 30 40 50

* Combining Array Element into String :-

It is possible in JavaScript

2 Function

1) concat()
separates each value with a comma.

2) join() :-
use a comma to separate values but you can specify a character other than comma to separate values.

```
<html>  
<body>  
  <script type="text/javascript">
```

```
    a = new Array();  
    a[0] = "Red";  
    a[1] = "Orange";  
    a[2] = "Yellow";  
    a[3] = "Green";  
    a[4] = "Blue";  
    a[5] = "Indigo";  
    a[6] = "Violet";  
    var str1 = a.join(",");  
    document.write(str1);  
    var str2 = a.concat();  
    document.write(str2);
```

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

For Eg: - join(",") → Red + Orange

* Changing elements of an array :-

There are various method

- 1) Shift
- 2) Push
- 3) Pop
- 4) Reverse

1) Shift :- This method removes the first element of an array.

Eg: -

```
<html>  
<body>
```

```
  <script type="text/javascript">  
    a = new Array();  
    a[0] = 10;  
    a[1] = 20;  
    a[2] = 30;  
    a[3] = 40;  
    a[4] = 50;  
    var num = a.shift();  
    document.write(num);  
    for (i = 0; i < a.length; i++)  
      document.write(a[i]);  
  </script>  
</body>  
</html>
```

1110

1120304050

2) Push Method :-

is used to create new element at the end of array

Eg: -

```
<html>
```

```
<body>
```

```
  <script type="text/javascript">  
    a = new Array();
```



```

a[0] = 10;
a[1] = 20;
a[2] = 30;
a[3] = 40;
a[4] = 50;

for (i=0, i < a.length, i++)
  document.write(a[i]);

a.push(60);
for (i=0, i < a.length, i++)
  document.write(a[i]);
// 10 20 30 40 50 60

```

3) Pop - This method returns & removes the last element of the array

```

Eg:-
<html>
<body>
<script type="text/javascript">
  a = new Array();
  a[0] = 10;
  a[1] = 20;
  a[2] = 30;
  a[3] = 40;
  a[4] = 50;
  for (i=0, i < a.length, i++)
    document.write(a[i]);

```

```

var val = a.pop();
document.write("The remove element "+val);
for (i=0, i < a.length, i++)
  document.write(a[i]);
</script>
</body>
</html>

```

4) Reverse Method:-

It is used to reverse the element present in array.

```

Eg:-
<html>
<body>
<script type="text/javascript">
  a = new Array();
  a[0] = 10;
  a[1] = 20;
  a[2] = 30;
  a[3] = 40;
  for (i=0, i < a.length, i++)
    document.write(a[i]);
  a.reverse();
  for (i=0, i < a.length, i++)
    document.write(a[i]);
// 40 30 20 10

```

Objects As Associative Array

Associative array is specialized array in which the elements are stored in (key, value) pair.

```

Eg:-
var a = {
  key: "one", value: 1,
  key: "two", value: 2,
  key: "three", value: 3;
};

<html>
<body>
<script type="text/javascript">
  a = new Object();
  a["one"] = 1;

```




Course & Semester :		Roll No. :	Date : / / 201	
Name of Subject :		Marks obtained :		
Sign. of Supervisor :	Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

0593

* Function

- we can define function anywhere in the script either in head or body section or in both.
- But it is a standard practice to define in head section & call that function from body section
- keyword function is used while defining the fn

Syntax:-

function function_name (arg1, arg2, ... argn)
 {
 statements;
 }

eg:-

```

<html>
<head>
<script type="text/javascript">
function Hello()
{
document.write("Hello in function");
}
</script> </head> <body>
<script type="text/javascript">
document.write("This statement is before
fn call");
Hello();
</script>
</body>
</html>

```

```

a["two"] = 2;
a["three"] = 3;
for (i in a)
document.write(i + " = " + a[i]);
</script>
</body>
</html>
o/p
One = 1
two = 2
three = 3

```


* Adding an Argument :-
we can pass some argument to the fn

Syntax :-

```
function func_name (arg1, arg2, ... argn)
{
  // body of fn
}
```

* Scope of Variable & Argument :-

- Scope is the block or area of program in which particulae variable or argument is accessible

- The scope is defined using two types of variables

① Local scope ② Global scope

1) Local scope :-

- If variable is defined inside fn then than variable is local variable & its scope is a local scope

- Local variable is accessible only within a fn in which it is defined. It is not accessible outside that fn.

eg:-

```
<html>
  <head>
    <script type="text/javascript">
      function A()
      {
        var a=100;
        document.write(a);
      }
    </script>
  </head>
</html>
```

function B()

```
{
  document.write(a); // output not
  // display bec
  // a is out of fn
  so
  </script>
  </head>
  <body> <script type="text/javascript">
    A();
    B();
  </script>
</body>
</html>
```

2) Global variable :-

It is defined outside the function, The variable having global scope is accessible by any fn

eg:- <html>

```
<head>
  <script type="text/javascript">
    var a=100;
    function A()
    {
      document.write(a);
    }
  </script>
</head>
```

```
<body>
  <script type="text/javascript">
    A();
    B();
  </script>
</body>
</html>
```


Calling function with argument:-
we can pass argument to function.

```
eg:- <html>
<head>
<script type="text/javascript">
function add(a,b)
{
c = a+b;
document.write("Addition is "+c);
}
</script>
</head>
<body>
<script type="text/javascript">
var x=10;
var y=20;
add(x,y);
</script>
</body>
</html>
```

Calling function without Argument:-
A function can also be called without passing any argument.
In this case, all the required variables are declared & used within that function.

```
eg:-
<html>
<head>
<script type="text/javascript">
function add()
{
var a=10;
var b=20;
}
```

* calling function from HTML:-
for calling function from HTML normally, JavaScript events are used.

```
eg:-
<html>
<head>
<script type="text/javascript">
function A()
{
alert("Inside the function A");
}
</script>
</head>
<body onload="A()">
</body>
</html>
```


Function calling another function :-
This is called nested fn

Eg:-

```

<html>
<head>
<script type="text/javascript">
function A()
{
    B();
}
function B()
{
    alert ("Inside B() via A()");
}
</script>
</head>
<body onload="A()">
</body>
</html>

```

Returning value from function :-

- using keyword return

This return value is either stored in variable or directly displayed on browser window.

Eg:-

```

<html>
<head>
<script type="text/javascript">
function A()

```

}

str = "I am function returned value";
return str;

```

</script>
</head>
<body>
<script type="text/javascript">
document.write(A());
</script>
</body>
</html>

```



String :-

It is collection of characters. The string is written within the quotes. either single or double quote.

Manipulating string :-

changing one to another string,
joining two string
changing the string from upper to lower case or from lower to upper

Methods

① concat(str)

- concatenates two string
eg:- str.concat(str2); will result in concatenation of string 1 & str2

② charAt(index_val)

- This method return the character specified by value index_val

③ substring(begin, end)

- return substring specified by begin & end character

④ toLowerCase()

- to convert all uppercase letters to lower case.

⑤ toUpperCase()

- to convert all lowercase letter to upper case

⑥ valueOf()

- return value of the string

* Property of string object is length

```
var s = "Hello";  
var l = s.length;
```

* Joining a string :- we can join two string using + operator

```
eg. <html> <script type="text/javascript">  
var f = "ABC";  
var s = "XYZ";  
var name = f+s;      ABCXZ  
alert(name);  
</script>  
</html>
```

Course & Semester :		Roll No. :	Date : / / 201	
Name of Subject :		Marks obtained :		
Sign. of Supervisor :		Sign. of Sub. Examiner :		
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks	

* Retrieving a character from given position

charAt() is method that returns the

character from specified index. The index of first character is 0. second character is 1 & so on.

In javascript characters in string are indexed from left to right. If no index is provided then default is 0.

eg:-

```
<html>  
<body>  
<script type="text/javascript">  
var a = "ABC"  
alert("The character at first position of string  
" + str + is " + str.charAt(0));      A  
</script>  
</body>  
</html>
```

* Retrieving a position of character in string.

- for finding the position of particular character we can use indexOf function.

- It returns the position of the first occurrence of specified value in string. This method is case sensitive

```
eg. <html>  
<body>
```



```

<script type="text/javascript">
var s = "I love India my country";
var n = str.indexof("India");
alert("The word 'India' is from index " +
n + " in the text " + s);
</script>
</body>
</html>

```

Retrieving a Position of character in string

Dividing Text :-

Any text is made up of two words. we can divide the text into collection of these words.

split() - divide the text
split() method creates new array & then copies portions of string, called substring, into its array element.

Syntax

var newArray = textname.split('delimiter char')

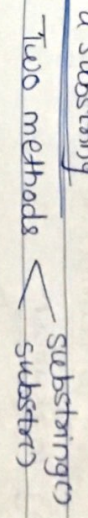
eg:-

```

<html>
<body>
<script type="text/javascript">
var Name = "Rajal Sanjay Durgamada";
var l = Name.split(' ');
alert("First Name " + l[0] + " \n Middle name " + l[1] + " \n Last Name " + l[2]);
</script>
</body>
</html>

```

* Copying a Substring



① substring()

syntax :- substring(start, end)

where start indicates the starting index & end indicates the ending index for extracting substring

eg:-

```

<html> <body> <script type="text/javascript">
var Name1 = "Rajal Sanjay Durgamada";
var Name2 = "Ankita";
var sub_str = Name1.substring(13, 21);
document.write(Name1);
// _____ (sub_str);
var Name3 = Name2 + " " + sub_str;
</script> </body> </html>

```

② substr

syntax :- substr(start, length)
to extract the substring from the text.

where start indicates the starting index & length indicates the no. of characters to be extracted

eg:-

```

<html> <body>
<script type="text/javascript">
var Name1 = "Rajal Sanjay Durgamada";
var Name2 = "Ankita";
var sub_str = Name1.substr(13, 21);
document.write(Name1);
// _____ (sub_str);
var Name3 = Name2 + " " + sub_str;
document.write(Name3);
</script> </body> </html>

```


* Converting String to Number & Number to String

1) String to Number
we have diff types of fn based on

type of numbers.
of string → integer (parseInt())
of string → float (parseFloat())

eg:-

```
<html>
<body>
<script type="text/javascript">
var s = "100";
var a = parseInt(s);
var b = 200; //100200
result = str + b;
document.write(result);
result = a + b; // 300
document.write(result);
</script>
</body>
</html>
```

2) Converting Number to String

using toString() - we can convert no to string

eg:-

```
<html> <body>
<script type="text/javascript">
var num = 100;
var str = num.toString();
document.write(str); // 100
</script>
</body>
</html>
```

* Changing Case of string

1) using toLowerCase(), we can convert the capital letters in the string to small letter

2) using toUpperCase(), we can convert the small letters to capital letters.

```
<html>
<body>
<script type="text/javascript">
var str1 = "WELCOME!";
var str2 = str1.toUpperCase();
document.write(str2);

var str3 = "welcome!";
var str4 = str3.toLowerCase();
document.write(str4);
</script>
</body>
</html>
```

* Finding Unicode of character

- computer can't understand characters. It understands only numbers.

- Hence whenever we type some letter, it get converted automatically to the number called unicode.

- The computer can't understand this uni- code number only.

- unicode is standard that assign number to every no, symbol, character that can be displayed on computer screen.

The ~~to~~ `charCodeAt()` is methods that returns unicode of a string.

for eg:-

```
<html> <body>
<script type="text/javascript">
  var ch='a';
  var n=ch.charCodeAt();
  document.write(n);           //97
</script>
</body>
</html>
```

We can obtain the letter or character from a unicode using the method `fromCharCode()`.

eg:-

```
<html>
<body>
<script type="text/javascript">
  var n=97;
  var ch=String.fromCharCode(n);
  document.write(ch)           // a
</script>
</body>
</html>
```