

## 2. Array, function & string



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

CLASS TEST I / II (201 - 201)

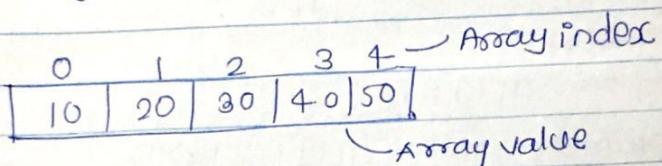
Course & Semester :	Roll No.:	Date : / /201	
Name of Subject :	Marks obtained :		
Sign. of Supervisor :	Sign. of Sub. Examiner :		
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

Supplied By: S.N.J.B.

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.

e.g:-

```
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++)
    b[i] = 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++)
    c[i] = i;
```

```
document.write(c[i]);
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Looping on Array :- Visiting each element present in array

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

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

```
Days = new Array();
Days[0] = "sunday";
```

```
for (i=0; Days.length; i++)
    document.write(Days[i]);
```

```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

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

```
<body>
```

```
<html>
```

# Adding an Array element

above code same for (10, 20, 30, 40, 50, 160, 70)

```
a.length = 80;
```

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

```
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.

a) 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>
```

```
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>
```

\* 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>
```

a) Push Method :-

is used to create new element at the end

of array

eg:-

```
<html>
<body>
<script type="text/javascript">
a = new Array();
a.join(" ") -> Red,Orange,---,Violet
for eg:- join(" ") → Red + Orange
```

#### 4) Reverse Method

$a[0] = 10;$   
 $a[1] = 00;$   
 $a[2] = 30;$   
 $a[3] = 40;$   
 $a[4] = 50;$

for ( $i=0; i < a.length; i++$ )  
    // 10 20 30 40 50

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

$\langle html \rangle$   
 $\langle body \rangle$   
 $\langle script type="text/javascript" \rangle$

$a = new Array();$   
 $a[0] = 10;$   
 $= 20;$

$= 30;$

$= 40;$

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

$a.reverse();$

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

In array.  
It is used to reverse the element present.

Eg:-

$\langle html \rangle$   
 $\langle body \rangle$   
 $\langle script type="text/javascript" \rangle$

$a = new Array();$

$= 10;$

$= 20;$

$= 30;$

$= 40;$

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

$a.reverse();$

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

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

for eg:-       $\langle key \rangle$        $\langle value \rangle$   
var a = { "one": 1, "two": 2, "three": 3 };

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

$\langle html \rangle$   
 $\langle body \rangle$

$\langle script type="text/javascript" \rangle$

$a = new Object();$   
 $a["one"] = 1;$   
 $\langle /script \rangle$   
 $\langle /body \rangle$   
 $\langle html \rangle$

Course & Semester:	Roll No.:	Date:
		/ 201

Name of Subject:	Sign. of Supervisor:	Sign. of Sub. Examiner:
Q. No. 1	Q. No. 2	Q. No. 3

Supplied By : S.N.J.B.	Marks obtained:
	Total Marks

```

a["two"] = 2;
a["three"] = 3;
for (i in a)
    document.write(i + "=" + a[i]);
    
```

### \* Function

0593

- 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;

3

eg:-

```

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

document.write ("Hello in function");

3

```

</script> </head> <body>
    
```

<script type="text/javascript">

document.write ("This statement is before

fn call");

Hello();

</script>

</body>

</html>

\* Adding an Argument :-

we can pass some argument to the fn

Syntax :-

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

c) \* Scope of Variable & Argument :-

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

- The scope is defined using two types of variables

① Local scope ② Global scope

D) 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 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);
    }

```

3

```
    A();
    B();

```

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

function B()

```
document.write(a); // output not displayed bcz a is out of B to go
```

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

A();

B();

```
</script>
```

```
</body>
```

```
</html>
```

Q) 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.getElementById("demo").innerHTML = a;
    }

```

```
</script> </head> </body>
```

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

```
        function B()
    
```

```
    {
        document.getElementById("demo").innerHTML = a;
    }

```

```
</script>
```

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

3

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">
add();
</script>
</body>
</html>
```

#### \* 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");
}

```

#### Calling function without Argument:-

A function can also be called without  
passing any argument.

In this case, all the required variable  
s are declared & used within that function.

eg:-

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

c = a+b;  
document.write("Addition is "+c);

}

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

```
add();
</script>
</body>
</html>
```

Function calling another function :-

This is called nested fn

Eg:-

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

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

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

J

### 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("s2"); will result in  
concatenation of string 1 & with 2

- (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

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; ABCXYZ

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

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

\* Receiving 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));
</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>
```

## \* Copying a Substring

Two methods substrings

### ① substr()

Syntax :- substr(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 = "Payal Sanjay Dungarwal";  
var Name2 = "Ankita";  
var substr = Name1.substr(13, 21);  
document.write(substr);  
</script> </body> </html>

Retrieving a Position of character in a 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 newstrArray = textname.split(delimiter char)

eg:-

<html>  
<body>

<script type="text/javascript">

var Name = "Payal Sanjay Dungarwal";

var L = Name.split(' ');\nalert("First Name "+L[0] + "\nMiddle\nname "+L[1] + "\nLast Name "+L[2]);

</script>

</body>

</html>

### ② substr

Syntax :- substr(start, length)

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

eg:-

<html> <body>

<script type="text/javascript">

var Name = "Payal Sanjay Dungarwal";

var Name1 = "Payal Sanjay Dungarwal";\nvar Name2 = "Ankita";\nvar substr = Name1.substr(13, 21);

document.write(substr);  
</script>

</body>

</html>

#### \* Changing case of string

- 1) using `toLowerCase()`, we can convert the capital letters in the string to small letters.
- 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>
```

#### 2) Converting Number to String :-

using `toString()` - we can convert no to string

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

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

```
var num=100;
```

```
var str=num.toString(); // 100
```

```
document.write(str); // 100
```

```
</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 ~~not~~ understand this unicode number only.

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

```
<html>
```

```
<body>
```

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

```
var s="100";
```

```
var a=parseInt(s);
```

```
var b=200; //100+200
```

```
result=s+b; // 1100
```

```
document.write(result);
```

```
document.write(result);
```

```
</script>
```

```
</body>
```

```
</html>
```

The `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>
```