

Unit II : Arrays, Functions & Strings

(Weightage - 14 marks)

(without option - 12 marks) (22)

(T/F-CO) - PRACTICALS

Unit-II Array, Function and String

- 2a. Create array to solve the given problem.
- 2b. Perform the specified string manipulation operation on the given String(s).
- 2c. Develop JavaScript to implement the given function.
- 2d. Develop JavaScript to convert the given Unicode to character form.
- 2e. Develop JavaScript to convert the given character to Unicode and vice-versa.

- 2.1 Array - declaring an Array, Initializing an Array, defining an Array elements, Looping an Array, Adding an Array element, sorting an Array element, Combining an Array elements into a String, changing elements of an Array, Objects as associative Arrays
- 2.2 Function – defining a function, writing a function, adding an arguments, scope of variable and arguments.
- 2.3 Calling a function – calling a function with or without an argument, calling function from HTML, function calling another function. Returning value from a function
- 2.4 String - manipulate a string, joining string, retrieving a character from given position, retrieving a position of character in a string, dividing text, copying a sub string, converting string to number and numbers to string, changing the case of string, finding a Unicode of a character-charCodeAt(), fromCharCode()

Array

The Array object lets you store multiple values in single variable.
It stores a fixed-size sequential collection of elements of same datatype.
An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of same type.

Defining an array

① By array literal

Syntax: var arrname = [value, value, ...];

② By creating instance of array

Syntax: var arrname = new Array();
new keyword is used to create instance
of array.

Initialising an Array

- * Initialisation is the process of assigning value of an array.
- * All the elements should be placed in parenthesis and separated by comma.

Example,

var a = [1, 2, 3, 4, 5]

var b = new Array ("C", "C++", "Java") ;

Defining an Array Elements

Array contains a list of elements; each element in the array is identified by its index.

0	1	2	3	4
1	2	3	4	5

array length = 5

first index = 0

last index = 4

- ↳ Assignment(=) operator is used to assign values to an array elements.

- ↳ Elements can be retrieving by index position.

My Visions:

<HTML>
<HEAD>

<TITLE> Array </TITLE>

<script type="text/javascript">

var a = [1, 2, 3, 4, 5];

document.write("Length of array = " + a.length);

</script>

</HEAD>

<body> </body>

</HTML>

[Output Length of array = 5]

Instance of array

<HTML>

<HEAD>

<TITLE> Array </TITLE>

<script type="text/javascript">

var a = new Array (3);

document.write("Length of array = " + length);

</script>

</head> <body> </body> </HTML>

) Output

Length of array = 3

You can add after defining size also

instance of array

<HTML>

<HEAD>

<TITLE> Defining array element <TITLE>

<SCRIPT type="text/javascript">

var d = new Array(5);

d[0] = "C";

d[1] = "C++";

d[2] = "JAVA";

d[3] = "VB";

d[4] = "PHP";

document.write("1st" + d[0]);

document.write("2nd" + d[1]);

document.write("3rd" + d[2]);

document.write("4th" + d[3]);

document.write("5th" + d[4]);

</SCRIPT> </HEAD> <BODY>

<Body> <HTML>

O/P

1st C

2nd C++

3rd JAVA

4th VB

5th PHP

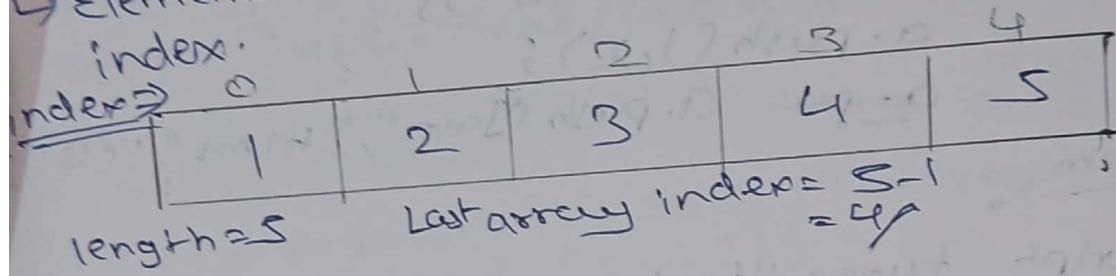
Traversing an Array

for Loop iterate over each item in an array.

Starting or first element is reference with an index of 0.

Ending or last element is array length minus 1.

Elements in array accessed by numeric index.



```
<script type="text/javascript">
var a = new Array(5);
a[0] = "C";
a[1] = "C++";
a[2] = "JAVA";
a[3] = "Python";
a[4] = "CSS";
for (var i=0; i < a.length; i++) {
    document.write("<br>Element = " + a[i]);
}
</script>
```

Output

Element = C.

Element = C++

Element = JAVA

Element = Python

Element = CSS

Adding an Array Element

↳ Using `push()` & `unshift()` methods

`push()`

It adds one or more elements at the end of array.

Syntax

`array.push(element1, element2, ..., elementn)`

Example

`a.push("Java");
a.push(1, 2);
a.push("Java", "HTML", "CSS");`

Program

```
<script type="text/javascript">  
var a = new Array();  
a[0] = "C";  
a[1] = "C++";  
document.write("Before : " + a.length + "  
a.push("JAVA");  
a.push("CSS", "Python");  
a.push(1, 2);  
document.write("After : " + a.length + "  
for (var i = 0; i < a.length; i++)  
    document.write("Elements : " + a[i] + "  
</script>
```

Output

Before : 2
After : 7

Element: C
Element: CSS
Element: Python
Element: 1
Element: 2

Unit III : Form and Event Handling

Unit III : It adds one or more elements at the beginning of array

Syntax - array.unshift(element1, ..., elementN)

Example - a.unshift("Java");
a.unshift(1,2);

Program

```
<script type="text/javascript">
    var a = new Array();
    a[0] = "JAVA";
    document.write("Before : " + a.length + "<br>");
    a.unshift("C", "C++");
    document.write("After : " + a.length + "<br>");
    for(var i=0; i < a.length; i++)
    {
        document.write("Element " + a[i]);
    }
</script>
```

O/P

Before = 1
After = 3

Element = C

Element = C++

Element = JAVA

Removing an Array Element

↳ Using `pop()` & `shift()` method

shift() : It removes first element of array.

Example: `a.shift();`

Syntax: `array.shift();`

<script>

```
var a = new Array("1", "2", "3", "4");
document.write("Before");
for(var i=0; i<a.length; i++)
{
    document.write("<br element">" + a[i]);
}
a.unshift();
document.write("<br> After <br>");
for(var i=0; i<a.length; i++)
{
    document.write("Element." + a[i]);
}
</script>
```

Output

Before
Element
Element 1
Element 2
Element 3
Element 4

After
Element 2
Element 3
Element 4

Unit III : Form and Event Handling

(Weightage - ~~10 marks~~)

- It removes last element of array

Syntax: array.pop()

Example: a.pop()

Before

1 | 2 | 3 | 4 | 5

a.pop();
Hr

After

1 | 2 | 3 | 4

Adding & Removing an Array Element

splice () method,

It add or remove elements from array

Syntax: array.splice(start, delete, element1, element2, ..., elementn)

Start: represent index from where the method to start to extract the element.

Delete: It is optional; no. of elements to be removed.

Element: to be inserted.

Example: a.splice(2, 0, "Java", "PHP");

Program

<script>



```
<HTML>
<HEAD>
    <TITLE> splice L1T1L2 </TITLE>
</HEAD>
<BODY>
<script type="text/javascript">
    var arr = new Array();
    arr[0] = "C";
    arr[1] = "A";
    document.write("Original length = " + arr.length);
    for (var i = 0; i < arr.length; i++) {
        document.write("<br>" + arr[i]);
    }
    arr.splice(2, 0, "S", "A");
    document.write("After " + arr.length);
    for (var i = 0; i < arr.length; i++) {
        document.write("<br>" + arr[i]);
    }
    arr.splice(2);
    document.write("After " + arr.length);
    for (var i = 0; i < arr.length; i++) {
        document.write("<br>" + arr[i]);
    }
</script>
```

Original length = 2
C
A

C
A
S
A
C
S
S
A
fter : 2

(weightage - 10 marks)

Sorting an Array Elements

↳ Using sort()

↳ Using reverse()

sort() method

It sorts the element of an array and returns the sorted array.

Syntax: `array.sort(compareFunction);`

Array will be sorted according to compare() function.

① compare function (function(a,b) {return a-b})
↳ ascending

② compare function (function(a,b) {return b-a})
↳ descending

✓ If result is negative a is sorted before b.

✓ If result is positive b is sorted before a.

✓ If result is 0 no change.

Program

Ascending

2 ways
a.sort() = [7|5|2|0|10] \Rightarrow [0|2|5|7|10]
(a.sort(function(a,b) {return a-b}))

Descending

a.sort(function(a,b) {return b-a}))

[2|10|7|5|2]

\Rightarrow [10|2|7|5|3|2]

reverse() method

It sorts the elements of array in reverse order. i.e descending order.

array.reverse();

Example

var r = a.sort(); \Rightarrow

a	1	2	3	4
---	---	---	---	---

 r.reverse(); \Rightarrow

1	2	3	4
---	---	---	---

5	4	3	2	1
---	---	---	---	---

Combining an Array Element into String

join()

concat()

join(): This function join all elements of an array into string

Syntax: array.join(separator)

Example: var r = a.join();

separator can be like /, ; - *
by default comma(,)

Programs

```
<script type='text/javascript'>
var a = new Array ("C", "C++", "JAVA");
document.write("Before join()");
for(var i = 0; i < a.length; i++) {
    document.write ("  
" + a[i]);
}
document.write ("After join()");
var r = a.join();
document.write (" " + r);
</script>
```

O/P

Before join()
C
C++
JAVA

After join()
C, C++, JAVA

cat()

This function is used to join two or more strings together in JavaScript.

This function returns a new string in combination of different strings passed to it as arguments.

Syntax: string1.concat(string2, string3...stringn)

For Example: var r = a.concat("VB");

Program

```
<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 concat "+r);
var z = a.concat("NET", "PHP");
document.write("<br> After joining = "+z);
</script>
```

Output

After concat = C, C++, JAVA, VB

After joining = C, C++, JAVA, , NET, PHP.

Object

- ↳ Associative arrays, are dynamic Object that the user redifines as needed.
- ↳ when you assigns values to keys in variables of array type, the array is transformed into an object, and it loses the attributes and methods of Array.
- ↳ we can create it assigning literal to a variable.

Syntax

```
array = {key1: "value1", key2: "value2",  
         key3: "value3"}
```

```
<script type="text/javascript">  
var a = {"Name": "Suraj", "Age": 23,  
         "Language": "English"};  
document.write("Name = "+a["Name"]);  
document.write("Age = "+a["Age"]);  
document.write("Language = "+a["Language"]);  
</script>
```

O/P

Name = Suraj Age = 23 Language = English

End Array

Functions

02 marks

A function is a subprogram designed to perform a particular task.

- ↳ Functions are executed when they are called.
This is known as invoking function.
- ↳ Values can be passed into functions and used within the function.
- ↳ Functions always return a value. In JS, if no return value is specified, the function will return undefined.
- ↳ Functions are Object.

Ddefining a Function

- ↳ The function keyword with Name of function.
- ↳ A list of parameters names etc enclosed in parenthesis.
- ↳ The statement enclosed in braces.
- ↳ The return keyword is (optional).

Syntax :

```
function name(parameters)  
{  
    statements  
}
```

Program

```
<script type="text/javascript">  
    function display()  
    {  
        document.write("Display Function");  
        display(); // call to function  
    }  
</script>
```

Arrays

Adding an Argument

Decl

(T)

Sy

(2)

Sy

- * when one or more variables declared within the parentheses of definition is known as arguments.
- * These arguments are used to hold for some task.

function name (parameters)

function add(no1, no2)

Program

<script>

var age = 20; // undefined as local globally

function display(name, age) // local

&

var name = "PYS";

document.write("Details = " + name + age);

y

display(); call

</script>

Output

Details:
PYS
Undefined

Global Scope

All the variables that you declare, is by default defined in global scope.

when a variable is outside function with a var.

Keyword then that variable global variable because it is available through all parts of script.

Local Scope

All the variables which are declared using var keyword within function then those variables are known as local variables and they are accessible with function only.

Calling a Function

Calling a Function from HTML

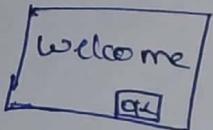
Functions can be called from HTML code in response of any particular event like page load, page unload etc.

Example

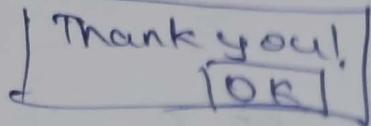
```
<HTML>
<Head>
</Head>
<Script type="text/javascript">
    function open()
    {
        alert("welcome");
    }
    function close()
    {
        alert("Thank you");
    }
</script>
<body onload="open()" onunload="close()">
</body>
</HTML>
```

Output

If page loads .



If page opens on other url



Answer =

Function calling another Function

↳ We call one function inside another function.

↳ The function which calls another function is called calling function.

↳ Function which is called by another function is called Refunction.

<script>

```
function accept(s)
```

```
{
```

```
document.write("Enter City = " + s);
```

```
}
```

```
function display()
```

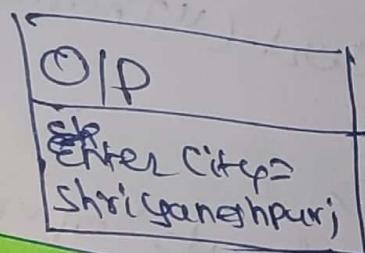
```
{
```

```
accept("Shri Ganeshpuri");
```

```
}
```

```
display();
```

</script>



Return a Value from Function

The return keyword stops the execution of function and value is returned from function to function caller.

Syntax

return value;

Program

<script>

```
function accept()
```

```
{
```

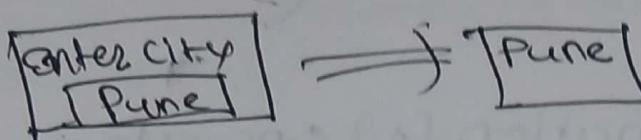
```
var s = prompt("Enter City");
```

```
}
```

```
return(s);
```

```
if(r2.accept())  
document.write(r);  
</script>
```

O/P



If return will be not there

undefined

String

- ↳ String are used for storing and manipulating text.
- ↳ String is zero or more characters written inside quotes.

Example, var c = "India"
var frame = "Bob";
var frame = new String("Bob");

* Manipulate a string

- ↳ String manipulation means operations on string.
Operations are :
 - ✓ concatenation
 - ✓ extraction of substring
 - ✓ finding length
 - ✓ finding position of character
 - ✓ changing case of string characters

Program

```
<script type="text/javascript">  
var c = "India";  
document.write("String value "+c);  
var s = new String("India");  
document.write ("As object: "+s);  
</script>
```

O/P

{ String value: India
As object: India }

Joining a String

1) String concatenation means joining two strings
a new string by placing the copy of second
string behind a copy of first string.

Methods

① Using concatenation (+) operator:

Syntax: string1 + string2

Example: var name = fname + sname;

② Using concat()

Syntax: concat(str1, str2...)

Example: var msg = "India"

var final = msg.concat(" is my country");

<script type="text/javascript">

var year = "TY";

var branch = "CO";

document.write("Value = " + (year + branch));

var msg = "India";

var result = msg.concat(" is my country");

document.write("value = " + result);

var fname = "Pranjal";

var sname = "Sare";

var fullname = fname.concat(sname);

document.write("value = " + fullname);

</script>

Output

Value = TY CO

Value = India is my country

Value = Pranjal Sare

Retrieving a character from given position

Character
charAt()

Returns the character at the "x" position
within the string.

Syntax: String.charAt(x)

Example: var mystr="Hello world";
mystr.charAt(2);

↳ Output = E

index=0
to last

Dividing text

split(): This function is used to split the given string into array of strings by separating it into substrings using a specified separator.

Syntax: string.split(separator, limit)
It means
char

```
var str="welcome to world";
document.write("Result = " + str.split("*", 3));
[ w * e * l ] .
```

Retrieving a Position of character in a String

↳ indexof()

↳ lastindexof()

↳ search()

[start] = its optional its between 0 or length of string

Substr is string (char) that we want to search and

IndexOf()

the function searches and return the index number of character or substring within string

Syntax: `IndexOf(substr, [start])`

`var mystr = "welcome world";`

`document.write("Position " + mystr.indexOf("l"))`

`0/P/E2`

`(1, 3) = 11`

LastIndexOf()

the function searches and return the index number last occurrence of characters or substring within string.

Searches string from end to beginning.

`var mystr = "welcome world";`

`mystr.lastIndexOf("l")`

`mystr.lastIndexOf("l", 3)`

`11`

`5`

Search()

the method searches the string for a specified value and return the position of match found

Syntax: `string.search(word)`

Example: `var str = "Hello world";`

`str = str.search("world");`

`11
8`

`search(10)`

`3`

(world)
at
= 1

Copying a Sub-string

substring() $\text{String.substring}(\text{index}, \text{index})$

- ↳ Returns the character in a string between "from" and "to" indexes.
- ↳ "to" is optional, and if omitted, up to the end of string is assumed.
- ↳ Syntax: `string.substring(from, [to])`

substr() $\text{String.substr}(\text{index} - \text{length}, \text{length})$

- ↳ Returns the character in a string beginning at "start" through the specified number of character, "length".
- ↳ "length" is optional, and if omitted, up to the end of string is assumed.

Syntax: `string.substr(from, [length])`

Program

```
<HTML>
<script>
var mystr="Welcome to js";
document.write("Result:" + mystr.substring(5));
document.write("Result: " + mystr.substring(5,10));
</script>
```

Output

Result: Welcome to js
Result: me to js

$\text{str} = \text{"Welcome Java"};$ $\text{str}.\text{substr}(5, 10))$	All ome . Java
---	-------------------

Converting String to Number & Number to String.

- ① parseInt() : converts to whole number
- ② parseFloat() : converts to decimal number
- ③ Number() : string to number. if not NaN output
- ④ toString() : This function used to convert number to string.

Changing the case of string

- ① toUpperCase() : characters will converted to uppercase.

```
var string1 = "India";  
string1.toUpperCase();  $\Rightarrow$  INDIA
```

- ② toLowerCase() : converted to lowercase

```
var string2 = "INDIA";  
string2.toLowerCase();  $\Rightarrow$  india
```

Finding the Unicode of character

What is Unicode?

The Unicode standard provide a unique number for every character, no matter what platform, device, application or language.

Example - 65 is English letter 'A' .

- 1. charCodeAt() : The function will return the unicode value of characters at specified position

Syntax: `string.charCodeAt(x);`

- 2. fromCharCode() : This method convert a given unicode number into a character.

Syntax : `String.fromCharCode(n1, n2, ..., x);`