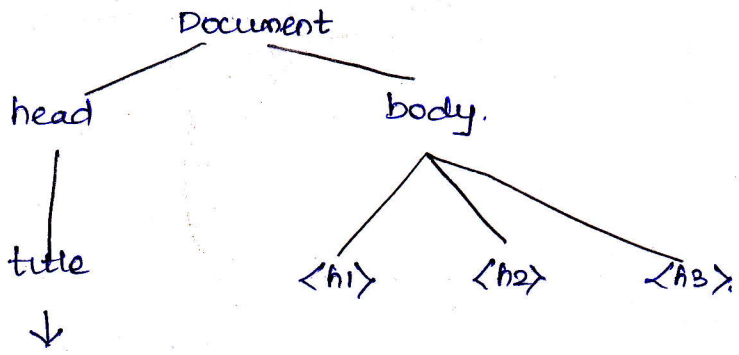
CSE - III YEAR - INTERNET PROGRAMMING.

UNIT II

1. Explain DOM in detail

The Document Object Modeling (DOM) is for defining the standard for accessing and manipulating HTML, XML and other scripting languages.

- Basically DOM is an Application Programming Interface (API) that defines the interface between HTML document and application program.



→ Every element in the DOM tree is called node.
→ The topmost single node in the DOM tree is called the root.
→ Every child node must have a parent node.
→ The bottommost nodes that have no children are called leaf nodes.
→ The nodes that have the common parent are called siblings.

ACCESSING ELEMENTS USING DOM:

a. Inserting an element using DOM:

```
<!DOCTYPE html>
<html> <body>
<div id = "div1">
<p id = "p1"> First Sentence </p1>
<pid = "p2"> Second Sentence </p2>
</div>
<script>
var pnode = document.createElement("p");
var node = document.createTextNode("Newly Added Sentence");
pnode.appendChild(node);
var element = document.getElementById("div1");
var nextnode = document.getElementById("p2");
element.insertBefore(pnode, nextnode);
</script> </Body> </html>
```

→ createElement("p") → function that is used to create new element/tag <p>.

→ createTextNode(" ") → Used to place the content inside this function.

→ nextnode = document.getElementById("p2") → Information in <p> tag with id p2 is accessed and stored in nextnode.

→ insertBefore(pnode, nextnode) → Appends/inserts the text content of pnode i.e., Newly added sentence before the content of nextnode.

Output will be:

First Sentence
Newly added Sentence
Second Sentence.

b. Removing an element using DOM:

- We can remove or delete particular from HTML document using DOM's removeChild method.

```
<!DOCTYPE html>
<html>
<body>
<div id = "div1">
<p id = "p1"> First Sentence </p1>
<p id= "p2"> Second Sentence </p2>
<p id = "p3"> Third Sentence</p3>
</div>
<script>
var parentnode = document.getElementById("div1");
var node = document.getElementById("p2");
parentnode.removeChild(node);
</script>
</body>
</html>
```

→ The variable parent node is assigned with root element div1.
→ The element with id p2 is assigned to the variable node.
→ Using removechild function it is deleted/removed.
→ getElementById() is used to fetch the element represented by the id

c. getElementByTagName()

```
<!DOCTYPE html>
<body>
<ul>
  <li> Coffee </li>
  <li> Tea </li>
</ul>
<p> Click the button </p>
<button onclick = "MyFunction()"> Try it </button>
<script>
function myfunction()
{
  var list = document.getElementsByTagName("UL")[0];
             list.getElementsByTagName("LI")[0].innerHTML = "Milk";
}
</script>
</body>
</html>
```

→ getElementsByTagName : used to fetch the element of UL tag and LI tag.

d. getElementByclassName().

```
<!DOCTYPE html>
<body>
<div id = "myDiv">
<p class = "child"> A p element with class = "child" in a div </p>
</div>
<p> Click the button </p>.
<button onclick = "myfunction()"> Try it </button>.
<p id = "demo"> </p>
<script>
function myfunction()
{
  var x = document.getElementById("myDiv").getElementByClassName
          ("child");
  document.getElementById("demo").innerHTML = x.length;
}
</script>
</body>
</html>
```

e. getElementByName:

```
<!DOCTYPE>
<html>
<body>
First Name : <input name = "fname" type = "text" value = "Michael"> <br/>
Last Name : <input name = "lname" type = "text" value = "Doug"> <br/>
<p> Click the button </p>
<button onclick = "myfunction()"> Try it </button>
<p id = "demo"> </p>
<script>
  function myfunction()
  {
     var x = documents.getElementsByName ("fname") [0]. tagName
     document.getElementById ("demo"). innerHTML = x;
  }
</script>
</body>
</html>.
```

---

2. FORM VALIDATION IN JAVASCRIPT.

Validation is a process of checking or validating the data that is given as input by the user.

Javascript program to validate a form consisting of Name, age, address, Email id, hobby (check box), Gender (radio box), country (Drop down menu).

```
<html>
<head>
<title> The student Registration Form </title>
<script type = text/javascript>
function validate()
{
    var i;
    var name_str = document. my_form. name;
    var phone_ID = document. my_form. ph_txt;
    var ph_str = document. my_form. ph_txt. value;
    var str = document. my_form. Email_txt. value;
    if ((name_str. value == null)) || (name_str. value == ""))
    {
      alert ("Enter some name")
      return false.
    }
```

```
if ((document.my_form.Age_txt.value < "5") && (document.my_form.
                                        age_txt.value > "21"))
{
   alert ("Invalid age")
   return false
}
if (ph_str.length < 1 || ph_str.length > 11)
{
   alert ("Invalid length of Phone Number")
   return false.
}
for (i=0; i < ph_str.length; i++)
{
   var ch = ph_str.charAt(i);
   if ((ch < "0") || (ch > "9"))
   {
      alert ("Invalid Phone number").
      phoneID.focus ()
      return false
   }
}
var emailID = document.my_form.Email_txt
if ((emailID.value == null) || (emailID.value == ""))
{
   alert ("Please Enter your Email ID")
   emailID.focus ()
   return false
}
if (document.my_form.group1[0].checked && !document.my_form.
                                        group1[0].checked)
{
   alert ("Select Gender");
   return false;
}
if (document.my_form.group1[0]
return true;
}
```

```
</script>
</head>
<body bgcolor = aqua>
<center><h3> Application Form </h3></center>
<form name = my_form onsubmit = validate()>
<strong> Name : </strong>
<input type = text name = name><br/>
<strong> Phone No: </strong>
<input type = text name = ph_txt><br/>
<strong> Email </strong>
<input type = text name = email_txt><br/>
<strong> Gender: </strong>
<input type = "radio" name = "group1" value = "Male"> Male
</input>
<input type = "radio" name = "group1" value = "Female"> Female
</input>
<strong> Hobby </strong>
<input type = "checkbox" name = "option1" value = "Sing">
Singing </input>
<input type = "checkbox" name = "option1" value = "Read">
Reading </input>
<br/><br/>.
<strong> Country: </strong>
<select name = "My-menu">
<option value = "India"> India </option>
<option value = "China"> China </option>
</select>
<center>
<input type = Submit value = "Submit"> <br/>
</center>
</body>
</html>.
```

3. Explain in detail about Event Handling with example program.

Event: Event is an activity that represents a change in the environment.

Eg: a) Mouse clicks
b) Pressing a particular key.

These kind of events are called Intrinsic events.

Event Handler: Event handler is a script that gets executed in response to these events.

Thus event handler enables the web document to respond the user activities through browser windows.

The following table displays the commonly used event handlers

S.No	Event	Event handlers	Description
1	blur	onBlur()	Invoked when any text/data/image gets blurred.
2	change	onChange()	Invoked when data in any HTML control field gets changed.
3	abort	onAbort()	Invoked whenever any process like loading page gets aborted.
4	click	onClick()	Invoked when any of the page element is clicked.
5	dbclick	onDbclick()	Invoked when any of the page element is double clicked.
6	dragdrop	onDragDrop()	Invoked whenever drag and drop option is performed.
7	focus	onFocus()	Invoked when any of the page element gain focus.
8	error	onError()	Invoked on occurrence of any javascript error.
9	keyup	onKeyUp()	Invoked as soon as the user releases the key.
10	keydown	onKeyDown()	Invoked as soon as the user press and release the key.
11	keypress	onKeyPress()	Invoked as soon as the user press the key.
12	mousemove	onMouseMove()	Invoked as soon as the user passes the mouse pointer on any text/object/image.
13	mousedown	onMouseDown()	Invoked as soon as the user presses the mouse key.
14	mouseout	onMouseout()	Invoked as soon as the mouse point is moved out of an object.
15	mouseup	onMouseup()	Invoked as soon as the user releases the mouse key.
16	resize	onResize()	Invoked as soon as any of the HTML elements is resized.

S.No	Event	Event Handlers	Description
17	Reset	onReset()	Invoked as soon as the user presses the reset button available on the web page.
18	Submit	onSubmit()	Invoked as soon as the user presses the submit button available on the web page.
19	select	onSelect()	Invoked whenever certain text of a given page element is selected.
20	unload	onUnload()	Invoked as soon as a given web document is closed.

a. onload()

```
<html>
<body onload = "myfunction()">
<h1> Hello World </h1>
<script>
function myfunction()
{
  alert (" Page is loaded");
}
</script>
</body>
</html>
```

b. onchange()

```
<html>
<head>
<script>
  function myfunction()
  {
    var x = document.getElementById ("fname");
    x.value = x.value.toUpperCase();
  }
</script>
</head>
<body>
Enter your name : <input type = "text" id = "fname"
                    onchange = "myfunction()">
</body>
</html>
```

c. onkeypress()

```
<html>
<body>
<p> A function for user pressing a key </p>
<input type = "text" onkeypress = "myfunction()">
<script>
function myfunction()
{
  alert ("You pressed a key");
}
</script>
</body>
</html>
```

d. onfocus().

```
<html>
<body>
<script>
  function myfunction(x)
  {
    x.style.background = "yellow";
  }
</script>
<body>
Enter your name : <input type = "text" onfocus = "myfunction()">.
</body>
</html>.
```

4. Explain objects in Java Script in detail.

In Javascript, object is a collection of properties. Javascript has the following properties.

1. Math object
2. Number object
3. Date object
4. Boolean object.
5. String objects.

A. Math object :

Used to perform the mathematical computations. Following are the methods available in Math object.

S.No	Method	Meaning	Code.
1	sqrt(num)	To find the square root.	<script type = "text/javascript"> var num = 100; document. write (Math. sqrt (num)); </script>.
2.	abs (num)	Returns absolute value of given number.	<script type = "text/javascript"> var x = -100; document. write (Math. abs (x)); </script>.
3.	ceil (num)	Returns the next positive integer value. Eg: ceil (10.3) gives 11.	<script type = "text/javascript"> var x = 10.5; document. write (Math. ceil (x)); </script>.
4.	floor (num)	Returns the integer part of the given number. Eg. floor (10.3) gives 10.	<script type = "text/javascript"> var x = 10.5; document. write (Math. floor (x)); </script>
5.	log (num)	Returns log value of num.	<script type = "text/javascript"> var x = 2; document. write (Math. log (x)); </script>.
6.	sin (num)	Returns the sin value of num	<script type = "text/javascript"> var x = 50; document. write (Math. sin (x));
	Note:	Similarly cos and	tan functions can be performed.

S.No	Method	Meaning	Code.
7.	pow(a,b)	Computes the pow a^b	<script type="text/javascript"> var a=5, b=2; document.write(Math.pow(a,b));
8.	min(a,b)	Returns the minimum among two numbers	</script> <script type="text/javascript"> var a=10, b=12 document.write(Math. min(a,b)).
	Similarly max of two numbers can be calculated.		

B. Number Object:

The properties of Number object are as follows:

S.No	Property	Meaning	Code.
1.	MAX_VALUE	Largest possible number is displayed.	<script type = "text/javascript"> document.write (Number.MAX_VALUE); </script>
2.	MIN_VALUE	Smallest possible number is displayed	<script type = "text/javascript"> document.write (Number.MIN_VALUE); </script>.
3.	NaN	When the input is not a number NAN is displayed.	-
4.	PI	Value of PI is displayed.	<script type = "text/javascript"> document.write (Number.PI); </script>.
5.	POSITIVE_INFINITY	The positive infinity gets displayed.	<Script type = "text/javascript"> document.write (Number.POSITIVE_INFINITY) </script>.
	Similarly NEGATIVE_INFINITY is executed.		

C. Date Object.

→ This object is used for obtaining the date and time.

→ This date and time value is based on computer's local time or it can be based on GMT.

The following are the commonly used methods of Date object.

S.No	Method	Meaning	Code

1.	getTime()	Returns the number in milliseconds. This value is the difference between current time and time value from 1st Jan 1970.	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString);
			document.write(my-date.getTime());
			</script>
			NOTE: FIRST 3 LINES OF CODE WILL BE COMMON FOR REMAINING FUNCTIONS.

2.	getDate()	Returns the current date.	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString());
			document.write(my-date.getDate();
			</script>.

3.	getDay()	Returns the current day as a number between 0 to 6 (i.e.) Sunday to Sat	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString());
			document.write(My-date.getDay();
			</script>

4.	getHours()	Returns the hour value ranging from 0 to 23.	<script type = "text/javascript">
			<var my-date = new Date();
			document.write(my-date.toString());
			document.write(my-date.getHours());
			</script>.

5.	getMilliSeconds()	Returns the milli-seconds ranging from 0 to 999.	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString());
			document.write(my-date.getMilliSeconds());
			</script>.

6.	getMinutes()	Returns the minute value ranging from 0 to 59.	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString());
			document.write(my-date.getMinutes());
			</script>.

7.	getSeconds()	Returns the second value from 0 to 59.	<script type = "text/javascript">
			var my-date = new Date();
			document.write(my-date.toString());
			document.write(my-date.getSeconds());
			</script>.

D. Boolean object: This object is the simplest kind of object which is used especially when we want to represent true and false values.

E. String Object: String is a collection of characters. The following are the methods of string. Assume S1 = "Hello"; S2 = "Parthi";

S.No	Method	Meaning	Code.
1.	concat (Str)	Concatenates two strings.	<script type = "text/javascript"> document.write (S1.concat(S2)); </script>
2.	length ()	Calculates the total number of characters in the given string.	<script type = "text/javascript"> document.write (S1.length()); </script>
3.	charAt(index-val)	Returns the character specified by the value index-val.	<script type = "text/javascript"> document.write (S1.charAt(1)); </script>
4.	substring (begin, end)	Returns the substring.	<script type = "text/javascript> document.write (S1.substring((1,3)); </script>
5.	tolowerCase ()	Converts upper case into lower case.	<script type = "text/javascript"> document.write (S1.tolowerCase); </script>.

5. Explain the concept of Exception Handling in detail.

Exception refers to run-time errors. Using exception handling mechanism, the errors occurring during run-time of a program can be successfully handled.

Try ... catch () Statement:

A block of statements that may cause exception will be written inside try block. If exception raises, the execution stops and catch block will handle the exception.

Throw statement:

The exceptions are thrown implicitly which are later caught by the catch construct provided in the code.

Example:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE = "text/javascript">
var x= prompt (" Enter a number between 0 and 5 :")
try {
```

```
if (x>5)
    throw error1
else if (x<0)
    throw error2
}
catch (e)
{ if (e == error1)
    alert (" Error ! You have entered a large value").
  if (e == error2)
    alert (" Error ! You value is very less")
}
</SCRIPT> </BODY> </HTML>.
```

---

6. What is a String function ? Write all methods of String function.

Strings are defined as series of characters enclosed within double quotation. Following are the predefined methods available in String.

Function	Description	Code.
toLowerCase()	Converts upper case to lowercase.	var name = " PRIYA "; document.write (name.toLowerCase());
toUpperCase()	Converts lower case to uppercase.	var name = "priya" document.write (name.toUpperCase());
concat()	Concate two strings.	var s1 = "Hi"; var s2 = "Hello"; document.write (s1.concat(s2));
charAt()	Returns the character whose index value is specified.	var s1 = " welcome"; document.write (s1.charAt(2));
charCodeAt()	Returns the unicode value of the character.	var s1 = " calcutta"; document.write (s1.charCodeAt(7)); 7th pos has a; a value is 97. o/p will be 97
substring(index, end)	Returns the substring.	var s1 = " welcome"; document.write (s1.substring(0,2)); o/p: wel.
indexOf()	Takes the character as argument and returns the numerical value.	var s1 = " welcome"; document.write (s1.indexOf ('e')); o/p : 1
lastIndexOf()	Same as indexOf() But performs searching from end.	var s1 = " welcome"; document.write (s1.lastIndexOf ('e')) o/p : 6.

7. Explain the Conditional statements and Looping statements in JavaScript.

Conditional Statements in JavaScript :-

The conditional statements are executed depending on the specified condition.

Conditional Statement	Description	Code

if	Executes if the condition is true	<script type = "text/javascript>
		var x = -2;
		if (x < 0)
		{
		document.write("x is a negative number
		}
		</script>.

if - else	If the condition is satisfied, statements inside if block is executed; otherwise statements under else block is executed.	<script type = "text/javascript">
		var a = 5, b = 10;
		if (a > b)
		document.write('A is greater');
		else
		document.write('B is greater');
		</script>.

if - else - if	Executed/used when we have more than two conditions.	<script type = "text/javascript">
		var a = 5, b = 6, c = 12;
		if ((a > b) && (a > c))
		document.write('A is greater')
		else if((b > a) && (b > c))
		document.write('B is greater')
		else
		document.write('C is greater'))
		</script>.

nested - if.	If condition inside another if condition is called as Java script.	<script type = "text/javascript">
		var marks = 90;
		if (marks > 50)
		{
		if (marks > 80)
		document.write("A Grade");
		else
		document.write("B Grade");
		}
		else
		document.write("Failed");
		</script>.

conditional statement	Description	code :

switch case.	Multiway decision making statement.	<script type = "text/javascript">.
		switch (new Date ().getDay ())
		{
		case 1 :
		day = "Sunday"
		break;
		case 2 :
		day = "Monday"
		break;
		case 3 :
		day = "Tuesday"; break;
		case 4:
		day = "Wednesday"; break;
		default: document.write ("Wrong day");
		}
		document.write (day);
		</script>.

Looping Statements :

Set of statements / statement will be repeated for a specified number of times until the condition is satisfied.

Looping statement	Code.	Output.

while
Top tested loop.
	<script type = "text/javascript">
	var i = 0;
	while (i<5)
	{ i = i+1;
	document.write (i)
	}
	</script>
		0
		1
		2
		3
		4

do-while
Bottom tested loop
	<script type = "text/javascript">
	var i = 0;
	do
	{
	i = i+1;
	document.write (i);
	}while (i<5);
	</script>
		0
		1
		2
		3
		4

for.
Used to iterate over elements.
	<script type = "text/javascript"> var i=0
	for (i=0; i<5; i++)
	{ document.write (i);
	}
		0
		1
		2
		3
		4

break.
Used to terminate the loop.
	<script type = "text/javascript">
	var i = 0;
	for (i = 10; i >= 0; i--)
	{ if (i == 5)
	break;
	}document.write (i)
		10
		9
		8
		7
		6

Continue statement skips the current iteration and the control goes to the beginning of the loop.

```
<script type = "text/javascript">
for (i = 10; i >= 0; i--)
{
    if (i == 5)
        continue;
    document.write (i).
}
```

Output will be : 10 9 8 7 6 4 3 2 1 0.

8. Explain the concept of arrays in JavaScript in detail.

→ An array contains a set of homogenous items in different languages.

→ Array can be declared as

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

→ Array can be initialised as follows:

```
var flowers = ["Rose", "Sunflower", "Lilly"];
```

Program:

```
<html>
<head>
<body>
<script language = "javascript">
var arr = [];
var books = ["B.Tech", "B.Sc", "Diploma"];
alert (books[1]);
</script>
</body>
</html>.
```

Array has many predefined functions. They are as follows.

P.T.O.

Function name	Code	Output.

concat()	<script type = "text/javascript">
	var a1 = ["sun", "mon", "tues"];
	var a2 = ["Jan", "Feb", "Mar"];
	var a3 = a1.concat(a2);
	document.write(a3);
	</script>
		sun, mon, tues, Jan, Feb, Mar.

join()	var array1 = ["sun", "mon", "tue"];
	document.getElementById("example").
	innerHTML = array1.join("-");
		sun-mon-tue.

pop() - Removes the last element.	var a1 = [10, 20, 30].
	a1.pop();
	document.write(a1);
		10, 20.

push() - Inserts an element at the end of the array.	var a1 = [10,20,30] a1.push(40); document.write(a1)	10, 20, 30, 40.
shift() Removes the first element of the array.	var a1 = [10, 20, 30] a1.shift() document.write(a1)	20, 30.
slice() Extracts a seqp of elements from the array.	var a1 = [10, 20, 30]; var a2 = a1.slice(1);	20, 30.

splice() Modifies the array by removing some array elements and adding new elements	var a1 = ["sun", "mon", "tue"];
	a1.splice(2,1,"thur");
	document.write(a1)
		sun, mon, thur.

sort() Sorts the elements of array.	var a1 = ["sun", "mon", "tue"];
	a1.sort();
	document.write(a1);
		mon, sun, tue.

Q. Explain how functions are defined in JavaScript. Also explain the parameter passing mechanism.

Function:

A function in JavaScript is a block of code which performs a specific task when it is called. Function accepts one or more arguments to perform a particular task.

Syntax:

```
function functionname (parameters)
{
  function body.
}.
```

Call By value:

```
<html>
 <head>
 <title> Adding two numbers </title>
 <body>
  <p> Addition of two numbers </p>
 <p id = "add" > </p>
 <script>
  var a = add (5,5);
  document. getElementById ("addition"). innerHTML = a;
  function add (x,y)
  {
    return x+y;
  }
 </script>
  </body>
  </html>.
```

Extra:

1. Regular Expression
2. Popup box – pg 4.96

Pgms.

1. Prime num – pg. 4.16
2. largest & smallest value among 10 elements – pg 4.19.
3. Pg 4.37. drawStrings.