WINTER – 2022 EXAMINATION

Subject Name: Client-Side Scripting **Model Ans** **Subject Code:** 22519

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**State the use of method in javascript with the help of suitable example.**	**2 M**

	Ans	A method/function is a set of statements that take inputs, do some specific computation, and produce output. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can call that function.
		
		Example:
		function Addition (number1, number2)
		{
		return number1 + number2;
		}
			Explanation - 1 M and Example- 1 M.

	b)	**List & Explain datatypes in JavaScript.**	**2 M**
	Ans	JavaScript provides different data types to hold different types of values. There are two types of data types in JavaScript, Primitive data type and Non-primitive data type **i) There are five types of primitive data types in JavaScript. They are as follows:** **String -** represents sequence of characters e.g., "hello" **Number -** represents numeric values e.g., 100	Any four, ½ for each

		Boolean - represents boolean value either false or true **Undefined -** represents undefined value **Null -** represents null i.e., no value at all **ii) The non-primitive data types are as follows:** **Object -** represents instance through which we can access members **Array -** represents group of similar values **RegExp -** represents regular expression	
	c)	**Write a simple calculator program using switch case in JavaScript.**	**2 M**

	Ans	<html>
		<body>
		<script>
		const number1 = parseFloat(prompt("Enter first number: "));
		const number2 = parseFloat(prompt("Enter second number: "));
		
		const operator = prompt("Enter operator (either +, -, *, / or %): ");
		let result;
		
		switch (operator) {
		case "+":
		result = number1 + number2;
		document.write(result);
		break;
		case "-":
		result = number1 - number2;
		document.write(result);
		break;
		
		case "*":
		result = number1 * number2;
		document.write(result);
		break;
		
		case "/":
		result = number1 / number2;
		document.write(result);
		break;
		
		case "%":
		result = number1 % number2;
		document.write(result);
		break;
			2 M for relevant program.

		default:
		document.write("Invalid operator");
		break;
		}
		</script>
		</body>
		</html>	

	d)	**Write a program using sort method of array object.**	**2 M**

	Ans	<html>
		<body>
		<script>
		var array =[5,1,9,7,5];
		// sorting the array
		sorted = array.sort();
		document.write(sorted);
		</script>
		</body>
		</html>
			2 M for relevant program.

	e)	**Describe property Getters & Setters.**	**2 M**

	Ans	JavaScript object accessors are used to access and update the objects. Getter and setter are used as object accessors to get or set object properties.
		
		Getter method helps in accessing the object methods as object properties.
		
		Setter method is used to set object properties.
		
		Using getter and setter the javascript provides better data security and data quality.
		
		Example:
		
		<!DOCTYPE html>
		<html>
		<body>
		<script>
		var car = {
		brand: "Toyota",
		color: "Blue",
		
		get getBrand () {
		return this.brand;
		},
		get getColor () {
		return this.color;
			1 m for each

		},
		
		set setBrand (newBrand) {
		this.brand = newBrand;
		},
		set setColor (newColor) {
		this.color = newColor;
		}
		};
		
		document.write("Car Brand: " + car.brand + "
		Car Color: " + car.color);
		
		car.setBrand = "Tesla";
		car.setColor = "Red";
		document.write("
		
		Car Brand: " + car.brand + "
		Car Color: " + car.color);
		</script>
		</body>
		</html>	

	f)	**Enlist & explain the use of any two Intrinsic JavaScript Functions.**	**2 M**
	Ans	An intrinsic function (or built-in function) is a function (subroutine) available for use in a given programming language whose implementation is handled specially by the compiler. You can use intrinsic functions to make reference to a data item whose value is derived automatically during execution. abs() - The ABS function returns the absolute value of the argument. sin() - The SIN function returns a numeric value that approximates the sine of the angle or arc specified by the argument in radians. sqrt() - The SQRT function returns a numeric value that approximates the square root of the argument specified. Date(): return current date. Len(): returns number of characters in the text. parseInt() - parseInt() function takes string as a parameter and converts it to integer. parseFloat() - parseFloat() function takes a string as parameter and parses it to a floating point number.	1 M for each function
	g)	**Describe browser location object.**	**2 M**

	Ans	i) The location object contains information about the current URL.
		ii) The location object is a property of the window object.
		iii) The location object is accessed with: window.location or just location.
		Example:
		<!DOCTYPE html>
		<html>
		<body>
		 The Window Location Object
		
		<script>
		let origin = window.location.origin;
		document.getElementById("demo").innerHTML = origin;
		</script>
		</body>
		</html>
			Explanation 1 M
			Example- 1 M

2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write a JavaScript program that will display current date in DD/MM/YYYY format.**	**4 M**
	Ans	<!DOCTYPE *html*> <html *lang*="en"> <head> <meta *charset*="UTF-8"> <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge"> <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <script> var d=new Date(); var currentDate=d.getDate()+'/'+(d.getMonth()+1)+'/'+d.getFullYear() document.write(currentDate)	Any relevant code 4 M.

```
    </script>

</body>

</html>
```

b) Write a JavaScript program that will remove the duplicate element from an array. **4 M**

Ans Any relevant code 4 M.

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <script>
      let arr = ["scale", "happy", "strength", "peace", "happy", "happy"];
      function removeDuplicates(arr) {
        let unique = [];
        for (i = 0; i < arr.length; i++) {
          if (unique.indexOf(arr[i]) === -1) {
            unique.push(arr[i]);
          }
        }
        return unique;
      }
      document.write(removeDuplicates(arr));
    </script>
  </body>
</html>
```

c) Write a JavaScript program that will display list of student in ascending order according to the marks & calculate the average performance of the class. **4 M**

Student Name	Marks
Amit	70
Sumit	78
Abhishek	71

Ans Any relevant code 4 M.

```
<html>

  <body>

    <script>

      var students = [["Amit", 70],["Sumit", 78],["Abhishek", 71],];

      var Avgmarks = 0;

      for (var i = 0; i < students.length; i++) {
```

```
      Avgmarks += students[i][1];
      for (var j = i + 1; j < students.length; j++) {
        if (students[i] > students[j]) {
          a = students[i];
          students[i] = students[j];
          students[j] = a
        }
      }
    }
    var avg = Avgmarks / students.length;
    document.write("Average grade: " + Avgmarks / students.length);
    document.write("<br><br>");
    for (i = 0; i < students.length; ++i){
      document.write(students[i]+"<br>")
    }
  </script>
 </body>
</html>
```

	d)	**Write and explain a string functions for converting string to number and number to string.**	**4 M**

	Ans	**To covert string to number** we can use parseInt() which converts a string number to a integer number. Similarly we can use parseFloat(), number() for converting string to number.
		
		Eg-
		var a=prompt('Enter a number');
		var b=parseInt(prompt('Enter a number'));
		document.write(typeof a+"
		");
		document.write(typeof b);
		To convert form number to string we can use toString()
		<html>
			Any relevant code with explanation 4 M.

```
<body>
<p>toString() returns a number as a string:</p>
<script>
let num = 12;
let text = num.toString();
document.write(num)
</script>
</body>
</html>
```

3. **Attempt any THREE of the following:** **12 M**

a) **Differentiate between concat( ) & join( ) methods of array object.** **4 M**

Ans Any 4 point=4M

concat()	join()
The concat() method concatenates (joins) two or more arrays. The concat() method returns a new array, containing the joined arrays.	The join() method returns an array as a string.
The concat() method separates each value with a comma only.	Any separator can be specified. The default is comma (,).
Syntax: *array1*.concat(*array2, array3, ..., arrayX*)	Syntax: *array*.join(*separator*)

Example:
<script>
const arr1 = ["CO", "IF"];
const arr2 = ["CM", "AI",4];
const arr = arr1.concat(arr1, arr2);
document.write(arr);
</script>
	Example:
	<script>
	var fruits = ["Banana", "Orange", "Apple", "Mango"];
	var text = fruits.join();
	document.write(text);
	var text1 = fruits.join("$$");
	document.write("
	"+text1);
	</script>

b) **Write a JavaScript function to check the first character of a string is uppercase or not.** **4 M**

Ans Correct function logic=4M

(any other relevant

```
<html>
<body>
<script>
function upper_case(str)
{
  regexp = /^[A-Z]/;
  if (regexp.test(str))
   {
     document.write("String's first character is uppercase");
```

		}
		else
		{
		document.write("String's first character is not uppercase");
		}
		}
		upper_case('Abcd');
		</script>
		</body>
		</html>
		
		OR
		<script>
		function firstIsUppercase(str)
		{
		if (str.length === 0)
		{
		return false;
		}
		
		return str.charAt(0).toUpperCase() === str.charAt(0);
		}
		
		if (firstIsUppercase(prompt("Enter text")))
		{
		document.write('First letter is uppercase');
		} else {
		document.write('First letter is NOT uppercase');
		}
		</script>
			logic can consider)

	c)	**Write a JavaScript function to merge two array & removes all duplicate values.**	**4 M**

	Ans	<html>
		<body>
		<script>
		function merge_array(array1, array2)
		{
		var result_array = [];
		var arr = array1.concat(array2);
		var len = arr.length;
		var assoc = {};
			Correct function logic=4 M
			
			(any other relevant logic can consider)

```
   while(len--)
    {
      var item = arr[len];

      if(!assoc[item])
      {
         result_array.unshift(item);
         assoc[item] = true;
      }
   }

   return result_array;
  }
var array1 = [1, 2, 3,4,7,9];
var array2 = [2, 30, 1,40,9];
document.write(merge_array(array1, array2));
</script>
</body>
</html>
```

Output:

3,4,7,2,30,1,40,9

OR

```
<html>
<body>
<script>
function mergearr(arr1, arr2)
 {
   // merge two arrays
   var arr = arr1.concat(arr2);
   var uniqueArr = [];
   // loop through array
   for(var i of arr) {
      if(uniqueArr.indexOf(i) === -1)
      {
         uniqueArr.push(i);
      }
   }
   document.write(uniqueArr);
}
```

		var array1 = [1, 2, 3,6,8];
		var array2 = [2, 3, 5,56,78,3]
		mergearr(array1, array2);
		</script>
		</body>
		</html>
		
		Output:
		1,2,3,6,8,5,56,78	

	d)	**Write a JavaScript function that will open new window when the user will clicks on the button.**	**4 M**

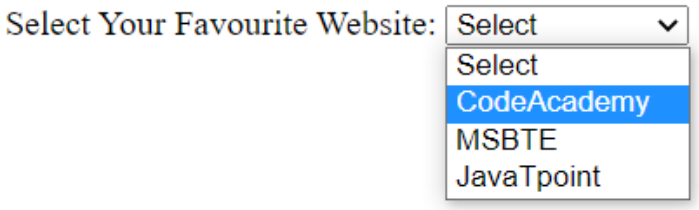
	Ans	<html>
		<body>
		<button onclick="openWin()">Open "New Window"</button>
		<script>
		var myWindow;
		function openWin()
		{
		myWindow = window.open("", "myWindow", "width=400,height=400");
		myWindow.document.write(" Hello Everyone.Welcome to new window. ");
		}
		</script>
		</body>
		</html>
			Correct function logic=4 M

4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Describe text Rollover with the help of example.**	**4 M**
	Ans	**Rollover means** a webpage changes when the user moves his or her mouse over an object on the page. It is often used in advertising. There are two ways to create rollover, using plain HTML or using a mixture of JavaScript and HTML. We will demonstrate the creation of rollovers using both methods. The keyword that is used to create rollover is the <onmousover> event. For example, we want to create a rollover text that appears in a text area. The text *"What is rollover?"* appears when the user place his or her mouse over the text area and the rollover text changes to *"Rollover means a webpage changes when the user moves his or her mouse over an object on the page"* when the user moves his or her mouse away from the text area.	Define Rollover-2 M Example-2 M (For example, any other relevant

		The HTML script is shown in the following example: Example: <html> <head></head> <Body> <textarea rows="2" cols="50" name="rollovertext" onmouseover="this.value='What is rollover?'" onmouseout="this.value='Rollover means a webpage changes when the user moves his or her mouse over an object on the page'"></textarea> </body> </html>	logic can be considered)
	b)	**Write a JavaScript program that will create pull-down menu with three options. Once the user will select the one of the options then user will redirected to that website.**	**4 M**

	Ans	<html>
		<head>
		<title>HTML Form</title>
		<script language="javascript" type="text/javascript">
		function getPage(choice)
		{
		page=choice.options[choice.selectedIndex].value;
		if(page != "")
		{
		window.location=page;
		}
		}
		</script>
		</head>
		<body>
		<form name="myform" action="" method="post">
		Select Your Favourite Website:
		<select name="MenuChoice" onchange="getPage(this)">
		<option value="select any option">Select</option>
		<option value="https://www.codecademy.com/catalog/language/javascript/">
		CodeAcademy </option>
		<option value="https://www.msbte.org.in">MSBTE</option>
		<option value="https://www.javatpoint.com/javascript-tutorial">JavaTpoint</option>
		</form>
		</body>
		</html>
			Creation of pull-down menus-1 M
			
			Correct function to redirect particular website-3 M
			
			OR
			
			any other relevant logic can be considered

Output:



c) Describe Quantifiers with the help of example. **4 M**

Ans

The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each special character has a specific connotation.
The +, *, ?, and $ flags all follow a character sequence.

Describe Quantifiers- 2 M

For Example-2 M

Sr.No.	Expression & Description
1	**p+** It matches any string containing one or more p's.
2	**p*** It matches any string containing zero or more p's.
3	**p?** It matches any string containing at most one p.(zero or one occurrences)
4	**p{N}** It matches any string containing a sequence of **N** p's
5	**p{2,3}** It matches any string containing a sequence of two or three p's.
6	**p{2, }** It matches any string containing a sequence of at least two p's.
7	**p$** It matches any string with p at the end of it.

Example:

```
<html>
<body>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction()
 {
  var str = "100, 1000 or 10000?";
  var patt1 = /\d{3,4}/g;
```

```
  var result = str.match(patt1);
  document.getElementById("demo").innerHTML = result;
}
</script>
 </body>
 </html>
```

d) Describe frameworks of JavaScript & its application. **4 M**

Ans

Frameworks of JavaScript:

Any 2

(1m for explanation and 1M for application)

1. ReactJs

React is based on a reusable component. Simply put, these are code blocks that can be classified as either classes or functions. Each component represents a specific part of a page, such as a logo, a button, or an input box. The parameters they use are called props, which stands for properties.

Applications:

React is a JavaScript library developed by Facebook which, among other things, was used to build Instagram.com.

2. Angular

Google operates this framework and is designed to use it to develop a Single Page Application (SPA). This development framework is known primarily because it gives developers the best conditions to combine JavaScript with HTML and CSS. Google operates this framework and is designed to use it to develop a Single Page Application (SPA). This development framework is known primarily because it gives developers the best conditions to combine JavaScript with HTML and CSS.

Applications:

Microsoft Office ,Gmail, Forbes, PayPal, Grasshopper, Samsung, Delta

3. Vue.js

Vue is an open-source JavaScript framework for creating a creative UI. The integration with Vue in projects using other JavaScript libraries is simplified because it is designed to be adaptable.

Application:

VueJS is primarily used **to build web interfaces and one-page applications**. It can also be applied to both desktop and mobile app development.

4. jQuery

It is a cross-platform JavaScript library designed to simplify HTML client-side scripting. You can use the jQuery API to handle, animate, and manipulate an event in an HTML document, also known as DOM. Also, jQuery is used with Angular and React App building tools.

Applications:

JQuery can be used to develop Ajax based applications.
It can be used to make code simple, concise and reusable.
It simplifies the process of traversal of HTML DOM tree.
It can also handle events, perform animation and add ajax support in web applications.

5. Node.js

Node.js is an open-source, server-side platform built on the Google Chrome JavaScript Engine. Node.js is an asynchronous, single-threaded, non-blocking I/O model that makes it lightweight and efficient.

Applications:

Paypal, LinkedIn, Yahoo, Mozilla, Netflix, Uber, Groupon, GoDaddy, eBay

e) Describe how to link banner advertisement to URL with example. — 4 M

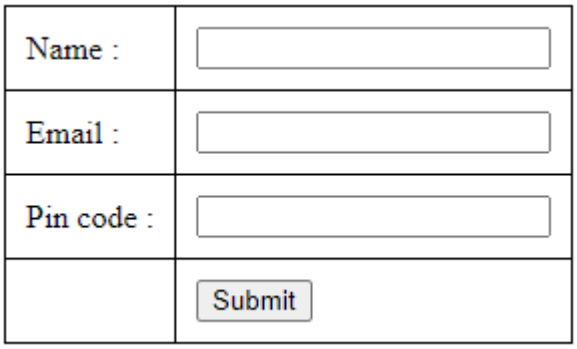
Ans (Banner-1 M, Example-3 M)

The banner advertisement is the hallmark of every commercial web page. It is typically positioned near the top of the web page, and its purpose is to get the visitor's attention by doing all sorts of clever things.

To get additional information, the visitor is expected to click the banner so that a new web page opens. You can link a banner advertisement to a web page by inserting a hyperlink into your web page that calls a JavaScript function rather than the URL of a web page. The JavaScript then determines the URL that is associated with the current banner and loads the web page that is associated with the URL.

Example:

```
<html>
<head>
<title>Link Banner Ads</title>
<script language="Javascript" type="text/javascript">
Banners = new Array('1.jpg','2.jpg','3.jpg')
BannerLink = new Array(
'google.com/','vpt.edu.in/', 'msbte.org.in/');
CurrentBanner = 0;
NumOfBanners = Banners.length;
function LinkBanner()
{
document.location.href =
"http://www." + BannerLink[CurrentBanner];
}
function DisplayBanners() {
if (document.images) {
CurrentBanner++
if (CurrentBanner == NumOfBanners) {
CurrentBanner = 0
}
document.RotateBanner.src= Banners[CurrentBanner]
setTimeout("DisplayBanners()",1000)
}
}
</script>
</head>
<body onload="DisplayBanners()" >
<center>
<a href="javascript: LinkBanner()"><img src="1.jpg"
```

		width="400" height="75" name="RotateBanner" /> </body> </html>	
5.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Write HTML script that will display following structure**  **Write the JavaScript code for below operations:** **(1) Name, Email & Pin Code should not be blank.** **(2) Pin Code must contain 6 digits & it should not be accept any characters.**	**6 M**

	Ans	<html>
		<head>
		<style>
		table,tr,td
		{
		border: solid black 1px;
		border-collapse: collapse;
		}
		
		td
		{
		padding: 10px;
		}
		</style>
		</head>
		<body>
		<table>
		<tbody>
		<tr>
		<td>Name : </td>
		<td> <input type="text" id="name" required></td>
		</tr>
		<tr>
		<td>Email : </td>
		<td> <input type="email" id="email" required></td>
		</tr>
		<tr>
			Creation of correct form and calling event-2 M
			
			Name, Email and Pin code should not be blank-2 M
			
			Pin code must contain 6 digits and it should not be blank-2 M

```
<td>Pin code : </td>
<td> <input type="number" id="pin" required></td>
</tr>
<tr>
<td></td>
<td><button onclick="submit()">Submit</button></td>
</tr>
</tbody>
</table>
</body>
<script>
function submit()
      {
         var name = document.getElementById("name").value;
         var email = document.getElementById("email").value;
         var pin = Number(document.getElementById("pin").value);
         if(name.length==0 || email.length==0 || pin.length==0)
         {
            alert("Please enter value in all fields.")
         }
         else
         {
            var pinpattern =/^[4]{1}[0-9]{5}$/;
            if( pinpattern.test(pin))
            {
               alert("Perfect Pin code");
            }
            else
            {
               alert("Wrong Pin code.");
            }
         }
      }
</script>
</html>
```

b) Write a webpage that displays a form that contains an input for user name and password. User is prompted to enter the input user name and password and password becomes the value of the cookie. Write the JavaScript function for storing the cookies. It gets executed when the password changes. — **6 M**

Ans

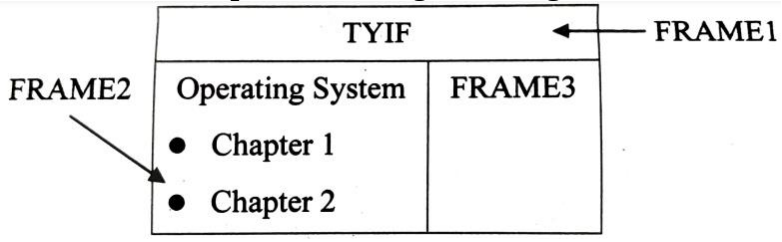
```
<html>
<head>
<script>
 function storeCookie()
 {
  var pwd = document.getElementById('pwd').value
  document.cookie = "Password=" + pwd + ";"
```

Creation of form=2 M

Storing and display cookie information- 4 M

```
 alert("Cookie Stored\n"+document.cookie);
 }
</script>
</head>
<body>
<form name="myForm">
Enter Username <input type="text" id="uname"/><br/>
Enter Password <input type="password" id="pwd"/><br/>
<input type="button" value="Submit" onclick="storeCookie()"/>
<p id="panel"></p>
</form>
</body>
</html>
```

c) **Write a JavaScript for creating following frame structure:** **6 M**



Chapter 1 and Chapter 2 are linked to the webpage Ch1.html and ch2.html respectively. When user click on these links corresponding data appears in FRAME3.

Ans

Step 1) create file frame1.html

```
<html>
<body>
<h1 align="center">FRAME1</h1>
</body>
</html>
```

Step 2) create frame2.html

```
<html>
<head>
<title>FRAME 2</title>
</head>
<body><H1>Operating System</H1>
<a href="Ch1.html" target="c"><UL>Chapter 1</UL></a>
<br>
<a href=" Ch2.html" target="c"><UL> Chapter 2</UL></a>
</body>
</html>
```

Correct frameset logic=6 M

OR
(any other relevant logic can be considered)

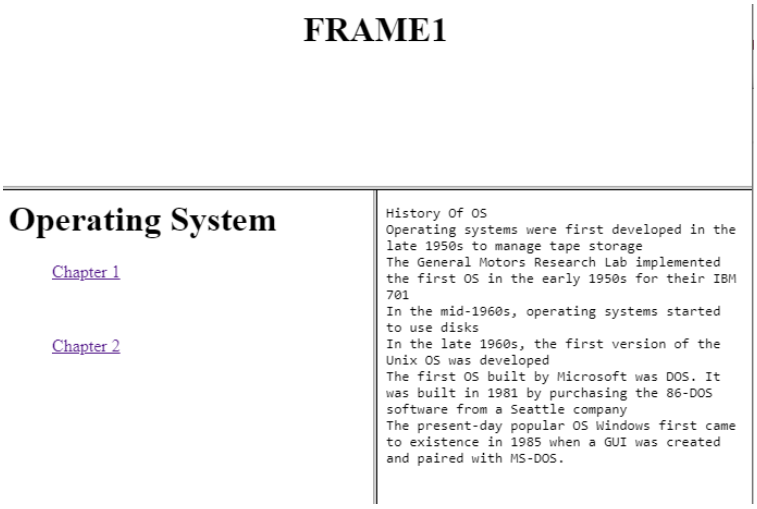
Step 3) create frame3.html

```
<html>
<body>
<h1>FRAME3</h1>
</body>
</html>
```

Step4) create frame_target.html

```
<html>
<head>
<title>Create a Frame</title>
</head>
<frameset rows="30%,*" border="1">
<frame src="frame1.html" name="a" />
<frameset cols="50%,*" border="1">
<frame src="frame2.html" name="b" />
<frame src="frame3.html" name="c" />
</frameset>
</frameset>
</html>
```

Output:



6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Write HTML script that will display dropdown list containing options such as Red, Green, Blue and Yellow. Write a JavaScript program such that when the user selects any options. It will change the background colour of webpage.**	**6 M**

	Ans	<html>
		<body>
		<label for="color">Choose a Background Color:</label>
		<select name="color" id="color" class="color" onchange="changeColor()">
		<option value="red">Red</option>
		<option value="green">Green</option>
		<option value="blue">Blue</option>
		<option value="yellow">Yellow</option>
		</select>
		<script type="text/javascript">
		function changeColor() {
		var color = document.getElementById("color").value;
		switch(color){
		case "green":
		document.body.style.backgroundColor = "green";
		break;
		case "red":
		document.body.style.backgroundColor = "red";
		break;
		case "blue":
		document.body.style.backgroundColor = "blue";
		break;
		case "yellow":
		document.body.style.backgroundColor = "yellow";
		break;
		default:
		document.body.style.backgroundColor = "white";
		break;
		}
		}
		</script>
		</body>
		</html>
			Creation of list-2 M
			
			Correct logic to Change background color as per selection-4 M

	b)	**Develop a JavaScript program to create Rotating Banner Ads.**	**6 M**

	Ans	<html >
		<head>
		<title>Banner Ads</title>
		<script>
		Banners = new Array('1.jpg','2.jpg','3.jpg');
		CurrentBanner = 0;
		function DisplayBanners()
		{
		if (document.images);
		{
		CurrentBanner++;
		if (CurrentBanner == Banners.length)
		{
		CurrentBanner = 0;
			Correct logic-6 M
			
			OR
			(any other relevant logic can be considered)

```
}
document.RotateBanner.src= Banners[CurrentBanner];
setTimeout("DisplayBanners()",1000);
}
}
</script>
</head>
<body onload="DisplayBanners()" >
<center>
<img src="1.jpg" width="400"
height="75" name="RotateBanner" />
</center>
</body>
</html>
```

c) Write a JavaScript for the folding tree menu. **6 M**

Ans

```
<html>
<head>
<style>
ul, #myUL {
  list-style-type: none;
}
.caret::before {
  content: "\25B6";
  color: black;
  display: inline-block;
  margin-right: 6px;
}

.caret-down::before {
  -ms-transform: rotate(90deg); /* IE 9 */
  -webkit-transform: rotate(90deg); /* Safari */'
  transform: rotate(90deg);
}

.nested {
  display: none;
}

.active {
  display: block;
}
</style>
</head>
<body>

<h2>Folding Tree Menu</h2>
<p>A tree menu represents a hierarchical view of information, where each item can have a number of subitems.</p>
```

Correct logic-6 M

OR
(any other relevant logic can be considered)

```
<p>Click on the arrow(s) to open or close the tree branches.</p>

<ul id="myUL">
  <li><span class="caret">India</span>
    <ul class="nested">
      <li>Karnataka</li>
      <li>Tamilnaadu</li>
      <li><span class="caret">Maharashtra</span>
        <ul class="nested">
          <li>Mumbai</li>
          <li>Pune</li>
          <li><span class="caret">Navi Mumbai</span>
            <ul class="nested">
              <li>Nerul</li>
              <li>Vashi</li>
              <li>Panvel</li>

            </ul>
          </li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
<script>
var toggler = document.getElementsByClassName("caret");
var i;
for (i = 0; i < toggler.length; i++) {
  toggler[i].addEventListener("click", function() {
    this.parentElement.querySelector(".nested").classList.toggle("active");
    this.classList.toggle("caret-down");
  });
}
</script>
</body>
</html>
```

WINTER – 19 EXAMINATIONS

Subject Name: Client Side Scripting **Model Answer** **Subject Code: 22519**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following :**	**10 M**
	a	**List any four features of Java script.**	**2 M**
	Ans	**Features of Java script** 1. JavaScript is a object-based scripting language. 2. Giving the user more control over the browser. 3. It Handling dates and time. 4. It Detecting the user's browser and OS, 5. It is light weighted. 6. Client – Side Technology 7. JavaScript is a scripting language and it is not java. 8. JavaScript is interpreter based scripting language. 9. JavaScript is case sensitive. 10. JavaScript is object based language as it provides predefined objects. 11. Every statement in javascript must be terminated with semicolon (;). 12. Most of the javascript control statements syntax is same as syntax of control statements in C language. 13. An important part of JavaScript is the ability to create new functions within scripts. Declare a function in JavaScript using **function** keyword.	**Any four features : ½ M each**
	b	**List the comparison operators in Java script.**	**2 M**

	Ans	**Comparison operators in Java script** = = Equal to != Not equal to > Greater than < Less than >= Greater than or equal to <= Less than or equal to = = = Equal value and equal type ! = = not equal value or not equal type	**Any 4 operators :1/2 M each**
	c	**Write a Java script to create person object with properties firstname, lastname, age, eyecolor, delete eyecolor property and display remaining properties of person object.**	**2 M**

	Ans	<html>
		<body>
		<script>
		var person = {
		firstname:"John",
		lastname:"Doe",
		age:50,
		eyecolor:"blue"
		};
		delete person.eyecolor; //delete person eyecolor
		document.write("After delete "+ person.firstname +" "+ person.lastname +" " +person.age +" "+ person.eyecolor);
		</script>
		</body>
		</html>
			Create person object : 1M
			
			Delete and display properties : 1M

	d	**Write a Java script that initializes an array called flowers with the names of 3 flowers. The script then displays array elements.**	**2 M**
	Ans	<html> <head>	**Initialization of array : 1M,**

		<title>Display Array Elements</title>
		</head>
		<body>
		<script>
		var flowers = new Array();
		flowers[0] = 'Rose ';
		flowers[1] = 'Mogra';
		flowers[2] = 'Hibiscus';
		for (var i = 0; i < flowers.length; i++)
		{
		document.write(flowers[i] + '
		');
		}
		</script>
		</body>
		</html>
			Display of array elements : 1M

	e	**Write Javascript to call function from HTML.**	**2 M**

	Ans	<html>
		<head>
		<title>Calling function from HTML</title>
		<script>
		function welcome()
		{
		alert("Welcome students");
		}
		function goodbye()
		{
		alert("Bye");
			Function declaration : 1M,
			
			Function call from HTML: 1M
			
			(Any other example allowed)

		} </script> </head> <body onload="welcome()" onunload="goodbye()"> </body> </html>	
	f	**Write a Javascript to design a form to accept values for user ID & password.**	**2 M**

	Ans	<html>
		<body>
		<form name="login">
		Enter Username<input type="text" name="userid">
		
		Enter Password<input type="password" name="pswrd">
		<input type="button" onclick="display()" value="Display">
		</form>
		<script language="javascript">
		function display()
		{
		document.write("User ID "+ login.userid.value + "Password : "+login.pswrd.value);
		}
		</script>
		</body>
		</html>
			Correct syntax: 1M,
			Correct logic: 1M

	g	**State any two properties and methods of location object.**	**2 M**
	Ans	**Properties of location object:** 1. hash 2. host 3. hostname 4. href 5. origin	**Any 2 properties : ½ M each**

		6. pathname 7. port 8. protocol 9. search **Methods of location object:** 1. assign( ) 2. reload( ) 3. replace( )	**Any 2 methods : ½ M each**
2		**Attempt any THREE of the following :**	**12 M**
	a	**Explain getter and setter properties in Java script with suitable example.**	**4 M**
	Ans	**Property getters and setters** 1. The accessor properties. They are essentially functions that work on getting and setting a value. 2. Accessor properties are represented by "getter" and "setter" methods. In an object literal they are denoted by get and set. **let obj = {** **get propName() {** **// getter, the code executed on getting obj.propName** **},** **set propName(value) {** **// setter, the code executed on setting obj.propName = value** **}** **};** 3. An object property is a name, a value and a set of attributes. The value may be replaced by one or two methods, known as setter and a getter. 4. When program queries the value of an accessor property, Javascript invoke getter method(passing no arguments). The retuen value of this method become the value of the property access expression. 5. When program sets the value of an accessor property. Javascript invoke the setter method, passing the value of right-hand side of assignment. This method is responsible for setting the property value. • If property has both getter and a setter method, it is read/write property. • If property has only a getter method , it is read-only property. • If property has only a setter method , it is a write-only property. 6. getter works when obj.propName is read, the setter – when it is assigned. **Example:** <html>	**Explanation : 2M**

		<head>
		<title>Functions</title>
		<body>
		<script language="Javascript">
		var myCar = {
		/* Data properties */
		defColor: "blue",
		defMake: "Toyota",
		
		/* Accessor properties (getters) */
		get color() {
		return this.defColor;
		},
		get make() {
		return this.defMake;
		},
		
		/* Accessor properties (setters) */
		set color(newColor) {
		this.defColor = newColor;
		},
		set make(newMake) {
		this.defMake = newMake;
		}
		};
		document.write("Car color:" + myCar.color + " Car Make: "+myCar.make)
		/* Calling the setter accessor properties */
		myCar.color = "red";
		myCar.make = "Audi";
		/* Checking the new values with the getter accessor properties */
		document.write("<p>Car color:" + myCar.color); // red
		document.write(" Car Make: "+myCar.make); //Audi
		</script>
		</head>
		</body>
		</html>
			Example : 2M
			
			(Any other example can be considered)

	b	**Explain prompt() and confirm() method of Java script with syntax and example.**	**4 M**
	Ans	**prompt()** The prompt () method displays a dialog box that prompts the visitor for input.The prompt () method returns the input value if the user clicks "OK". If the user clicks "cancel" the method returns null. **Syntax: window.prompt (text, defaultText)**	**For Each explanation/** **syntax : 1M,** **Example : 1M**

Example:

```
<html>
    <script type="text/javascript">
    function msg(){
        var v= prompt("Who are you?");
        alert("I am "+v);
    }
    </script>
    <input type="button" value="click" onclick="msg()"/>
</html>
```

(Any other example can be considered)

confirm()

It displays the confirm dialog box. It has message with ok and cancel buttons. Returns Boolean indicating which button was pressed

Syntax:

window.confirm("sometext");

Example :

```
<html>
<script type="text/javascript">
function msg(){
    var v= confirm("Are u sure?");
    if(v==true){
    alert("ok");
}
else{
    alert("cancel");
}
}
</script>
<input type="button" value="delete record" onclick="msg()"/>
</html>
```

	c	**Write a Java script program which computes, the average marks of the following students then, this average is used to determine the corresponding grade.** **Student Name** | **Marks** **Sumit** | **80** **Kalpesh** | **77** **Amit** | **88** **Tejas** | **93** **Abhishek** | **65** **The grades are computed as follows :** **Range** | **Grade** **<60** | **E** **<70** | **D** **<80** | **C** **<90** | **B** **<100** | **A**	**4 M**

	Ans	<html>
		<head>
		<title>Compute the average marks and grade</title>
		</head>
		<body>
		<script>
		var students = [['Summit', 80], ['Kalpesh', 77], ['Amit', 88], ['Tejas', 93], ['Abhishek', 65]];
		var Avgmarks = 0;
		for (var i=0; i < students.length; i++) {
		Avgmarks += students[i][1];
		}
		var avg = (Avgmarks/students.length);
		document.write("Average grade: " + (Avgmarks)/students.length);
		document.write("
		");
		if (avg < 60){
			Correct logic : 2M,
			Correct Syntax: 2M
			(any other logic can be considered)

		document.write("Grade : E");
		}
		else if (avg < 70) {
		document.write("Grade : D");
		}
		else if (avg < 80) {
		document.write("Grade : C");
		} else if (avg < 90) {
		document.write("Grade : B");
		} else if (avg < 100) {
		document.write("Grade : A");
		}
		</script>
		</body>
		</html>
		Output (Optional)
		Average grade: 80.6
		Grade : B	

	d	**Write the use of chatAt() and indexof() with syntax and example.**	**4 M**
	Ans	**charAt()** The charAt() method requires one argument i.e is the index of the character that you want to copy. **Syntax:** var SingleCharacter = NameOfStringObject.charAt(index); **Example:** var FirstName = 'Bob';	**Each syntax : 1M,** **Example : 1M**

var Character = FirstName.charAt(0); //o/p B

indexOf()

The indexOf() method returns the index of the character passed to it as an argument.

If the character is not in the string, this method returns –1.

Syntax:

var indexValue = string.indexOf('character');

Example:

var FirstName = 'Bob';

var IndexValue = FirstName.indexOf('o'); //o/p index as 1

3 **Attempt any THREE of the following :** **12 M**

a **Differentiate between concat() and join() methods of array object.** **4 M**

Ans

concat()	**join()**
Array elements can be combined by using concat() method of Array object.	Array elements can be combined by using join() method of Array object.
The concat() method separates each value with a comma.	The join() method also uses a comma to separate values, but you can specify a character other than a comma to separate values.
Eg: var str = cars.concat() The value of str is 'BMW, Audi, Maruti'	Eg: var str = cars.join(' ') The value of str in this case is 'BMW Audi Maruti'

b **Write a JavaScript that will replace following specified value with another value in string.** **4 M**

		String = "I will fail" **Replace "fail" by "pass"**	

	Ans	<html>
		<head>
		<body>
		<script>
		var myStr = 'I will fail';
		var newStr = myStr.replace(fail, "pass");
		document.write(newStr);
		</script>
		</body>
		</head>
		</html>
			Correct program with any proper logic: 4 M

	c	**Write a Java Script code to display 5 elements of array in sorted order.**	**4 M**
	Ans	<html> <head> <title> Array</title> </head> <body> <script> var arr1 = ["Red", "red", "Blue", "Green"] document.write("Before sorting arra1=" + arr1); document.write(" After sorting arra1=" + arr1.sort()); </script> </body> </html>	**Correct program with any proper logic : 4 M**
	d	**Explain open() method of window object with syntax and example.**	**4 M**

	Ans	The open() method of window object is used to open a new window and loads the document specified by a given URL.
		
		MyWindow = window.open()
		
		The open() method returns a reference to the new window, which is assigned to the MyWindow variable. You then use this reference any time that you want to do something with the window while your JavaScript runs.
		A window has many properties, such as its width, height, content, and name—to mention a few. You set these attributes when you create the window by passing them as parameters to the open() method:
		
		• The first parameter is the full or relative URL of the web page that will appear in the new window.
		• The second parameter is the name that you assign to the window.
		• The third parameter is a string that contains the style of the window.
		
		We want to open a new window that has a height and a width of 250 pixels and displays an advertisement that is an image. All other styles are turned off.
		
		Syntax:
		
		MyWindow = window.open('webpage1.html', 'myAdWin', 'status=0, toolbar=0, location=0, menubar=0, directories=0, resizable=0, height=250, width=250')
		
		Example:
		
		<html >
		<head>
		<title>Open New Window</title>
		<script >
		function OpenNewWindow() {
		MyWindow = window.open('webpage1.html', 'myAdWin', 'status=0, toolbar=0, location=0,
		menubar=0, directories=0, resizable=0, height=250, width=250')
		}
		</script>
		</head>
		<body>
		<FORM action=" " method="post">
		<INPUT name="OpenWindow" value="Open Window" type="button" onclick="OpenNewWindow()"/>
		</FORM>
		</body>
		</html>
			Explanation: 1 M
			
			syntax: 1 M
			
			Example: 2 M
			
			(Any other example can be considered)

4		**Attempt any THREE of the following :**	**12 M**
	a	**Describe regular expression. Explain search () method used in regular expression with suitable example.**	**4 M**

	Ans	**Regular Expression:**
		A regular expression is very similar to a mathematical expression, except a regular expression tells the browser how to manipulate text rather than numbers by using special symbols as operators.
		
		Search() method:
		str.search() method takes a regular expression/pattern as argument and search for the specified regular expression in the string. This method returns the index where the match found.
		
		Example:
		
		<html>
		<body>
		<script>
		function myFunction() {
		
		// input string
		var str = "Good Morning!";
		
		// searching string with modifier i
		var n = str.search(/Morning/i);
		
		document.write(n + '
		');
		
		// searching string without modifier i
		var n = str.search(/Morning/);
		
		document.write(n);
		}
		myFunction();
		</script>
		</body>
		</html>
			Regular Expression: 1 M
			
			search() method: 1 M
			
			Example: 2M

	b	**List ways of protecting your web page and describe any one of them.**	**4 M**
	Ans	**Ways of protecting Web Page:** 1)Hiding your source code 2)Disabling the right MouseButton 3) Hiding JavaScript 4) Concealing E-mail address.	**List: 1 M** **Explanation any one: 3M**

1) Hiding your source code

The source code for your web page—including your JavaScript—is stored in the *cache*, the part of computer memory where the browser stores web pages that were requested by the visitor. A sophisticated visitor can access the cache and thereby gain access to the web page source code.
However, you can place obstacles in the way of a potential peeker. First, you can disable use of the right mouse button on your site so the visitor can't access the View Source menu option on the context menu. This hides both your HTML code and your JavaScript from the visitor.
Nevertheless, the visitor can still use the View menu's Source option to display your source code. In addition, you can store your JavaScript on your web server instead of building it into your web page. The browser calls the JavaScript from the web server when it is needed by your web page.
Using this method, the JavaScript isn't visible to the visitor, even if the visitor views the source code for the web page.

2)Disabling the right MouseButton

The following example shows you how to disable the visitor's right mouse button while the browser displays your web page. All the action occurs in the JavaScript that is defined in the <head> tag of the web page.

The JavaScript begins by defining the BreakInDetected() function. This function is called any time the visitor clicks the right mouse button while the web page is displayed. It displays a security violation message in a dialog box whenever a visitor clicks the right mouse button
The BreakInDetected() function is called if the visitor clicks any button other than the left mouse button.

Example:

```
<html>
<head>
<title>Lockout Right Mouse Button</title>
<script language=JavaScript>

function BreakInDetected(){
alert('Security Violation')
return false
}
function NetscapeBrowser(e){
if (document.layers||
document.getElementById&&!document.all){
if (e.which==2||e.which==3){
```

```
BreakInDetected()
return false
}
}
}
function InternetExploreBrowser(){
if (event.button==2){
BreakInDetected()
return false
}
}
if (document.layers){
document.captureEvents(Event.MOUSEDOWN)
document.onmousedown=NetscapeBrowser()
}
else if (document.all&&!document.getElementById){
document.onmousedown=InternetExploreBrowser()
}
document.oncontextmenu=new Function(
"BreakInDetected();return false")

</script>
</head>
<body>
<table width="100%" border=0>
<tbody>
<tr vAlign=top>
<td width=50>
<a>
<ing height=92 src="rose.jpg"
width=70 border=0
onmouseover="src='rose1.jpg'"
onmouseout="src='rose.jpg'">
</a>
</td>
<td>
<img height=1 src="" width=10>
</td>
<td>
<a>
<cTypeface:Bold><u> Rose Flower</U></b>
</a>
</font><font face="arial, helvetica, sans-serif"
size=-1><BR>Rose Flower
```

```
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

3) Hiding JavaScript

You can hide your JavaScript from a visitor by storing it in an external fi le on your web server. The external fi le should have the .js fi le extension. The browser then calls the external file whenever the browser encounters a JavaScript element in the web page. If you look at the source code for the web page, you'll see reference to the external .js fi le, but you won't see the source code for the JavaScript.
The next example shows how to create and use an external JavaScript file. First you must tell the browser that the content of the JavaScript is located in an external
file on the web server rather than built into the web page. You do this by assigning the fi le name that contains the JavaScripts to the src attribute of the <script> tag, as shown here:

```
<script src="MyJavaScripts.js"
language="Javascript" type="text/javascript">
```

Next, you need to defi ne empty functions for each function that you define in the external JavaScript fi le.

```
<html >
<head>
<title>Using External JavaScript File</title>
<script src="myJavaScript.js" language="Javascript" type="text/javascript">

function OpenNewWindow(book) {
}
</script>
</head>
<body>
<tablewidth="100%" border=0>
<tbody>
<tr vAlign=top>
<td width=50>
<a>
<img height=92 src="rose.jpg" width=70 border=0 name='cover'>
</a>
```

```
</td>
<td>
<img height=1 src="" width=10>
</td>
<td>
<a onmouseover="OpenNewWindow(1)" onmouseout="MyWindow.close()">
<b><u>Rose </u></b>
</a>
<br>
<a onmouseover="OpenNewWindow(2)" onmouseout="MyWindow.close()">
<b><u>Sunflower</U></b>
</a>
<br>
<A onmouseover="OpenNewWindow(3)" onmouseout="MyWindow.close()">
<b><u>Jasmine </u></b>
</a>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

The final step is to create the external JavaScript fi le. You do this by placing all function definitions into a new fi le and then saving the fi le using the .js extension.

MyJavaScript.js file:

```
function OpenNewWindow(book) {
if (book== 1)
{
document.cover.src='rose.jpg'
MyWindow = window.open('', 'myAdWin', 'titlebar=0 status=0, toolbar=0, location=0, menubar=0, directories=0, resizable=0, height=50, width=150,left=500,top=400')
MyWindow.document.write( 'Rose flower')
}
if (book== 2)
{
document.cover.src='sunflower.jpeg'
MyWindow = window.open('', 'myAdWin', 'titlebar=0 status=0, toolbar=0, location=0, menubar=0, directories=0, resizable=0, height=50, width=150,left=500,top=500')
MyWindow.document.write( 'sunflower flower')
}
```

```
if (book== 3)
{
document.cover.src='jasmine.gif'
MyWindow = window.open('', 'myAdWin', 'titlebar=0
status=0, toolbar=0, location=0, menubar=0, directories=0, resizable=0, height=50,
width=150,left=500,top=600')
MyWindow.document.write( 'Jasmine Flower')
}
}
```

After you create the external JavaScript fi le, defi ne empty functions for each function that is contained in the external JavaScript fi le, and reference the external
JavaScript fi le in the src attribute of the <script> tag, you're all set.

4) Concealing E-mail address:

Many of us have endured spam at some point and have probably blamed every merchant we ever patronized for selling our e-mail address to spammers. While e-mail addresses are commodities, it's likely that we ourselves are the culprits who invited spammers to steal our e-mail addresses.
Here's what happens: Some spammers create programs called *bots* that surf the Net looking for e-mail addresses that are embedded into web pages, such as those placed there by developers to enable visitors to contact them. The bots then strip these e-mail addresses from the web page and store them for use in a spam attack. This technique places developers between a rock and a hard place. If they place their e-mail addresses on the web page, they might get slammed by spammers. If they don't display their e-mail addresses, visitors will not be able to get in touch with the developers.
The solution to this common problem is to conceal your e-mail address in the source code of your web page so that bots can't fi nd it but so that it still appears on the web page. Typically, bots identify e-mail addresses in two ways: by the mailto: attribute that tells the browser the e-mail address to use when the visitor wants to respond to the web page, and by the @ sign that is required of all e-mail addresses. Your job is to confuse the bots by using a JavaScript to generate the e-mail address dynamically. However, you'll still need to conceal the e-mail address in your JavaScript, unless the JavaScript is contained in an external JavaScript file, because a bot can easily recognize the mailto: attribute and the @ sign in a JavaScript.
Bots can also easily recognize when an external fi le is referenced. To conceal an e-mail address, you need to create strings that contain part of the e-mail address and then build a JavaScript that assembles those strings into the e-mail address, which is then written to the web page.
The following example illustrates one of many ways to conceal an e-mail address.

It also shows you how to write the subject line of the e-mail. We begin by creating four strings:

• The first string contains the addressee and the domain along with symbols &, *, and _ (underscore) to confuse the bot.
• The second and third strings contain portions of the mailto: attribute name. Remember that the bot is likely looking for mailto:.
• The fourth string contains the subject line. As you'll recall from your HTML training, you can generate the TO, CC, BCC, subject, and body of an e-mail from within a web page.
You then use these four strings to build the e-mail address. This process starts by using the replace() method of the string object to replace the & with the @ sign and the * with a period (.). The underscores are replaced with nothing, which is the
same as simply removing the underscores from the string.
All the strings are then concatenated and assigned to the variable b, which is then assigned the location attribute of the window object. This calls the e-mail program on the visitor's computer and populates the TO and Subject lines with the strings generated by the JavaScript.

```
<html>
<head>
<title>Conceal Email Address</title>
<script>
function CreateEmailAddress(){
var x = manish*c_o_m'
var y = 'mai'
var z = 'lto'
var s = '?subject=Customer Inquiry'
x = x.replace('&','@')
x = x.replace('*','.')
x = x.replace('_','')
x = x.replace('_','')
var b = y + z +':'+ x + s
window.location=b
}
-->
</script>
</head>
<body>
<input type="button" value="Help"
onclick="CreateEmailAddress()">
</body>
</html>
```

	c	**Create a slideshow with the group of three images, also simulate next and previous transition between slides in your Java Script.**	**4 M**

	Ans	<html>
		<head>
		<script>
		pics = new Array('1.jpg' , '2.jpg' , '3.jpg');
		count = 0;
		function slideshow(status)
		{
		if (document.images)
		{
		count = count + status;
		if (count > (pics.length - 1))
		{
		count = 0;
		}
		if (count < 0)
		{
		count = pics.length - 1;
		}
		documet.imag1.src = pics[count];
		}
		}
		</script>
		</head>
		<body>
		<img src="1.jpg" width="200" name="img1">
		
		<input type="button" value="Next" onclick="slideshow(1)">
		<input type="button" value="Back" onclick="slideshow(-1)">
		</body>
		</html>
			Correct program: 4 M
			
			(Any other example can be considered)

	d	**Explain text rollover with suitable example.**	**4 M**
	Ans	You create a rollover for text by using the onmouseover attribute of the <A> tag, which is the *anchor* tag. You assign the action to the onmouseover attribute the same way as you do with an <IMG> tag. Let's start a rollover project that displays a flower titles. Additional information about a flower can be displayed when the user rolls the mouse cursor over the flower name. In this example, the image of the flower is displayed. However, you could replace the flower image with an advertisement or another message that you want to show about the flower. <html> <head> <title>Rollover Text</title>	**Explanation: 2 M** **Program: 2 M** **(Any other example can be considered)**

		</head> <body> <TABLE width="100%" border="0"> <TBODY> <TR vAlign="top"> <TD width="50"> <a> <IMG height="92" src="rose.jpg" width="70" border="0" name="cover"> </TD> <TD> <IMG height="1" src="" width="10"> </TD> <TD> <A onmouseover= "document.cover.src='sunflower.jpg'"> <B><U>Sunflower</U></B> </A> <A onmouseover= "document.cover.src='jasmine.jpg'"> <B><U>Jasmine</U></B> </A> <A onmouseover= "document.cover.src='rose.jpg'"> <B><U>Rose</U></B> </A> </TD> </TR> </TBODY> </TABLE> </body> </html>	
	e	**Write a Java script to modify the status bar using on MouseOver and on MouseOut with links. When the user moves his mouse over the links, it will display "MSBTE" in the status bar. When the user moves his mouse away from the link the status bar will display nothing.**	**4 M**
	Ans	<html> <head> <title>JavaScript Status Bar</title></head> <body> 	**Correct program: 4 M**

		MSBTE </body> </html>	
5		**Attempt any TWO of the following :**	**12 M**
	a	**Write a HTML script which displays 2 radio buttons to the users for fruits and vegetables and 1 option list. When user select fruits radio button option list should present only fruits names to the user & when user select vegetable radio button option list should present only vegetable names to the user.**	**6 M**

	Ans	<html>
		<head>
		<title>HTML Form</title>
		<script language="javascript" type="text/javascript">
		function updateList(ElementValue)
		{
		with(document.forms.myform)
		{
		if(ElementValue == 1)
		{
		optionList[0].text="Mango";
		optionList[0].value=1;
		optionList[1].text="Banana";
		optionList[1].value=2;
		optionList[2].text="Apple";
		optionList[2].value=3;
		}
		if(ElementValue == 2)
		{
		optionList[0].text="Potato";
		optionList[0].value=1;
		optionList[1].text="Cabbage";
		optionList[1].value=2;
		optionList[2].text="Onion";
		optionList[2].value=3;
		}
		}
		}
		</script>
		</head>
		<body>
		<form name="myform" action="" method="post">
		<p>
			Correct script code: 4M HTML code: 2M

		<select name="optionList" size="2"> <option value=1>Mango <option value=2>Banana <option value=3>Apple </select> <input type="radio" name="grp1" value=1 checked="true" onclick="updateList(this.value)">Fruits <input type="radio" name="grp1" value=2 onclick="updateList(this.value)">Vegetables <input name="Reset" value="Reset" type="reset"> </p> </form> </body> </html>	
	b	**Describe, how to read cookie value and write a cookie value. Explain with example.**	**6 M**
	Ans	Web Browsers and Servers use HTTP protocol to communicate and HTTP is a stateless protocol. But for a commercial website, it is required to maintain session information among different pages. For example, one user registration ends after completing many pages. But how to maintain users' session information across all the web pages. Cookies are a plain text data record of 5 variable-length fields − • **Expires** − The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser. • **Domain** − The domain name of your site. • **Path** − The path to the directory or web page that set the cookie. This may be blank if you want to retrieve the cookie from any directory or page. • **Secure** − If this field contains the word "secure", then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists. • **Name=Value** − Cookies are set and retrieved in the form of key-value pairs Cookies were originally designed for CGI programming. The data contained in a cookie is automatically transmitted between the web browser and the web server, so CGI scripts on the server can read and write cookie values that are stored on the client.	**Reading cookie with example: 3M** **Writing cookie with example: 3M** ****Note: Combined of both code is also acceptable**

JavaScript can also manipulate cookies using the **cookie** property of the **Document** object. JavaScript can read, create, modify, and delete the cookies that apply to the current web page.

Storing Cookies

The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this.

```
document.cookie = "key1 = value1;key2 = value2;expires = date";
```

Here the **expires** attribute is optional. If you provide this attribute with a valid date or time, then the cookie will expire on a given date or time and thereafter, the cookies' value will not be accessible.

```
<html>
  <head>
    <script type = "text/javascript">
      <!--
        function WriteCookie()
        {
          if( document.myform.customer.value == "" ) {
            alert("Enter some value!");
            return;
          }
          cookievalue = escape(document.myform.customer.value) + ";";
          document.cookie="name=" + cookievalue;
          document.write ("Setting Cookies : " + "name=" + cookievalue );
        }
      //-->
    </script>
  </head>
  <body>
    <form name = "myform" action = "">
      Enter name: <input type = "text" name = "customer"/>
      <input type = "button" value = "Set Cookie" onclick = "WriteCookie();"/>
    </form>
  </body>
</html>
```

Reading Cookies

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie. The document.cookie string will keep a list of name=value pairs separated by semicolons, where **name** is the name of a cookie and value is its string value.

You can use strings' **split()** function to break a string into key and values as follows:-

```
<html>
  <head>
   <script type = "text/javascript">
      <!--
        function ReadCookie()
        {
          var allcookies = document.cookie;
          document.write ("All Cookies : " + allcookies )
          // Get all the cookies pairs in an array
          cookiearray = allcookies.split(';');

          // Now take key value pair out of this array
          for(var i=0; i<cookiearray.length; i++) {
            name = cookiearray[i].split('=')[0];
            value = cookiearray[i].split('=')[1];
            document.write ("Key is : " + name + " and Value is : " + value);
          }
        }
      //-->
    </script>

  </head>
  <body>

    <form name = "myform" action = "">
      <p> click the following button and see the result:</p>
      <input type = "button" value = "Get Cookie" onclick =
    "ReadCookie()"/>
    </form>
```

		</body> </html>	
	c	**Write a java script that displays textboxes for accepting name & email ID & a submit button. Write java script code such that when the user clicks on submit button** **(1) Name Validation** **(2) Email ID Validation.**	**6 M**

	Ans	<html>
		<head>
		<title>Form Validation</title>
		</head>
		<body>
		<form action = "/cgi-bin/test.cgi" name = "myForm" onsubmit = "return(validate());">
		<table cellspacing = "2" cellpadding = "2" border = "1">
		<tr>
		<td align = "right">Name</td>
		<td><input type = "text" name = "Name" /></td>
		</tr>
		
		<tr>
		<td align = "right">EMail</td>
		<td><input type = "text" name = "EMail" /></td>
		</tr>
		
		<tr>
		<td align = "right"></td>
		<td><input type = "submit" value = "Submit" /></td>
		</tr>
		</table>
		</form>
		</body>
		</html>
		<script type = "text/javascript">
		<!--
		// Form validation code will come here.
		function validate() {
		if(document.myForm.Name.value == "") {
		alert("Please provide your name!");
		document.myForm.Name.focus() ;
		return false;
		}
		if(document.myForm.EMail.value == "") {
			Correct Html code: 2M
			
			Correct Script code: 4M
			
			(Any other example can be considered)

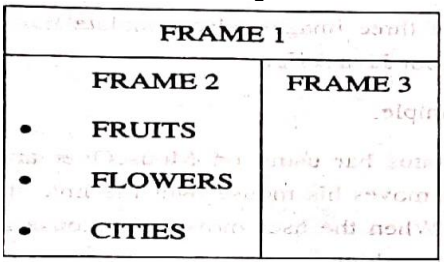
```
            alert( "Please provide your Email!" );
            document.myForm.EMail.focus() ;
            return false;
          }
          var emailID = document.myForm.EMail.value;
          atpos = emailID.indexOf("@");
          dotpos = emailID.lastIndexOf(".");

          if (atpos < 1 || ( dotpos - atpos < 2 )) {
            alert("Please enter correct email ID")
            document.myForm.EMail.focus() ;
            return false;
          }
           return( true );
        }
      //-->
</script>
```

6		**Attempt any TWO of the following :**	**12 M**
	a	**Describe how to evaluate checkbox selection. Explain with suitable example.**	**6 M**
	Ans	Evaluating Checkbox Selection:	**Correct Explanation : 3M & Correct Example:3M (Any other example can be considered)**

- A checkbox is created by using the input element with the type=”checkbox” attribute-value pair.
- A checkbox in a form has only two states(checked or un-checked) and is independent of the state of other checkboxes in the form. Check boxes can be grouped together under a common name.
- You can write javascript function that evaluates whether or not a check box was selected and then processes the result according to the needs of your application.
- Following example make use of five checkboxes to provide five options to the user regarding fruit.

```
<html>
        <head>
         <title>HTML Form</title>
         <script language="javascript" type="text/javascript">
                function selection()
                {
                        var x ="You selected: ";
                        with(document.forms.myform)
                        {
                                if(a.checked == true)
                                {
                                        x+= a.value+ " ";
```

```
                                }
                                if(b.checked == true)
                                {
                                        x+= b.value+ " ";
                                }
                                if(o.checked == true)
                                {
                                        x+= o.value+ " ";
                                }
                                if(p.checked == true)
                                {
                                        x+= p.value+ " ";
                                }
                                if(g.checked == true)
                                {
                                        x+= g.value+ " ";
                                }
                                document.write(x);
                        }
                }
         </script>
        </head>
        <body>
                <form name="myform" action="" method="post">
                  Select Your Favourite Fruits: <br>
                  <input type="checkbox" name="a" value="Apple">Apple
                  <input type="checkbox" name="b" value="Banana">Banana
                  <input type="checkbox" name="o" value="Orange">Orange
                  <input type="checkbox" name="p" value="Pear">Pear
                  <input type="checkbox" name="g" value="Grapes">Grapes
                  <input type="reset" value="Show" onclick="selection()">
                </form>
        </body>
</html>

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

	b	**Write a script for creating following frame structure**  **FRUITS, FLOWERS AND CITIES are links to the webpage fruits.html, flowers.html, cities.html respectively. When these links are clicked corresponding data appears in FRAME 3.**	**6 M**
	Ans	<html> <head> <title>Frame Demo</title> </head> <body> <table border="1"> <tr> <td align="center" colspan="2"> FRAME 1 </td> </tr> <tr> <td> FRAME 2 FRUITS FLOWERS CITIES </td> <td> FRAME 3 <iframe name="mainframe"></iframe> </td> </tr> </table> </body> </html>	**Frame part: 2M for each**

	c	**Write a javascript to create a pull-down menu with three options [Google, MSBTE, Yahoo] once the user will select one of the options then user will be redirected to that site.**	**6 M**

	Ans	<html>
		<head>
		<title>HTML Form</title>
		<script language="javascript" type="text/javascript">
		function getPage(choice)
		{
		page=choice.options[choice.selectedIndex].value;
		if(page != "")
		{
		window.location=page;
		}
		}
		</script>
		</head>
		<body>
		<form name="myform" action="" method="post">
		Select Your Favourite Website:
		<select name="MenuChoice" onchange="getPage(this)">
		<option value="https://www.google.com">Google</option>
		<option value="https://www.msbte.org.in">MSBTE</option>
		<option value="https://www.yahoo.com">Yahoo</option>
		</form>
		</body>
		</html>
			pull-down menu code: 2M each

SUMMER – 2024 EXAMINATION
MODEL ANSWER

Subject: Client Side Scripting Language **Subject Code:** **22519**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No	Sub Q.N.	Answer	Marking Scheme
1.	a) Ans.	**Attempt any FIVE of the following:** **Write features of JavaScript** 1. It is an object-based scripting language. 2. It gives the user more control over the browser. 3. It is light weighted. 4. Client – Side Technology 5. JavaScript is interpreter based scripting language. 6. JavaScript is case sensitive. 7. JavaScript is object based language as it provides predefined objects.	**10** **2M** ***Any 2 features 1M each***
	b)	**List and describe any four methods of Math object**	**2M**

	Ans.	Math. round(value)- It returns value rounded to its nearest integer. Math.ceil(value)- It returns value rounded up to its nearest integer. Math.floor(value)- It returns value rounded down to its nearest integer. Math.trunc(Value)- It returns value as integer part of value. Math.pow (number, power)- It returns value as power of specified number. Math.sqrt(value)- It returns square root of value. Math.abs(value)- It returns absolute –positive value for given value. Math.min ()- It returns lowest value in a list of values. Math.max ()- It returns highest value in a list of values.	***Any 4 methods 1/2M each***

	c)
	Ans.
		Write a JavaScript program that will print even numbers from 1 to 20
		Note: Any other relevant logic shall be considered
		<html>
		<body>
		<script type="text/javascript">
		vari;
		for(i=1; i<=20;i++)
		{
		if(i%2==0)
		{
		document.write(i+"
		");
		}
		}
		</script>
		</body>
		</html>
			2M
			Correct program 2M

	d) **Ans.**	**Write a JavaScript program to display the elements of array in ascending and descending order.** ***Note: Any other relevant logic shall be considered*** <script> function func() {	**2M** ***Correct program 2M***

```
        let arr = [45,12,32,78]
        document.write("Original Array="+arr);
        document.write("<br>Sorted Array="+arr.sort());
         document.write("<br>Reverse Array="+arr.reverse());
}
func();
</script>
```

e) **Give syntax of and explain function in JavaScript with suitable example** — **2M**

Ans.

Function is a collection of one or more statements written to execute a specific task.

Explanation 1M

Syntax 1M

Syntax to define a function:

```
function function_name([Arguments])
{
        Statement block;
        [return statement;]
}
```

Example:

```
function display ( )
{
alert ("WELCOME TO JAVASCRIPT");
}
```

f) **Enlist and explain any two mouse events.** — **2M**

Ans.

onclickevent: This event occurs when a mouse button is clicked on or over a form element.
Example:<input type="text" onclick=" function ()">

ondblclickevent: This event occurs when a mouse button is double clicked on or over a form element.
Example:<input type="text" ondblclick=" function ()">

onmousedownevent: This event executes when a mouse button is clicked while cursor is over an element.
Example:<input type="text" onmousedown=" function ()">

Any two mouse events with explanation 1M each

		onmouseupevent: This event executes when a mouse button is released while the cursor is over an element. Example:<input type=”text” onmouseup=” function ()”> **onmouseoverevent**: This event executes when mouse cursor moves onto an element. Example:<input type=”text” onmouseover=” function ()”> **onmousemoveevent**: This event executes when mouse cursor is moved while over an element. Example:<input type=”text” onmousemove=” function ()”> **onmouseoutevent:** This event executes when mouse cursor is moved away from an element. Example:<input type=”text” onmouseout=” function ()”>	
	g) Ans.	**Explain the term JavaScript URL** A **URL** (Uniform Resource Locator) is the address of a unique resource on the internet. It is one of the key mechanisms used by browsers to retrieve published resources, such as HTML pages, CSS documents, images, and so on. **Examples of URLs:** https://developer.mozilla.org/en-US/docs/Learn/ https://developer.mozilla.org/en-US/search?q=URL	**2M** ***Correct explanation 2M***
2.	**a) Ans.**	**Attempt any <u>THREE</u> of the following:** **State the use of Object, Method and Property in JavaScript** **Object:** In JavaScript, almost everything is an object. In JavaScript, an object is used to represent standalone entity, with properties and type. Each object has its unique identity based on fields, buttons, interface elements, etc. For example, two forms placed on web page can have different elements and interface with respect to their use. So, each form can have unique name or id that can be referenced by JavaScript.	**12** **4M** ***Correct use of all terms 4M***

		Property: A property is a value that is associated with an object. The properties of an object are used to define the characteristics of the object. You access the properties of an object with a simple dot-notation. For example, A form object in a web page can have properties like width, height, etc. **Method:** A method is used to define a function associated with an object to perform a specific task. Methods are defined the way normal functions are defined, except that they have to be assigned as the property of an object. For example, A submit button placed on a form is an object. Clicking on submit button causes the button to process a method i.e. when a click event occurs an action is performed and method executes.	
	b) **Ans.**	**Explain setter and getter properties in JavaScript with the help of suitable example** **Property getters and setters** 1. The accessor properties. They are essentially functions that work on getting and setting a value. 2. Accessor properties are represented by "getter" and "setter" methods. In an object literal they are denoted by get and set. get –It is used to define a getter method to get the property value set –It is used to define a setter method to set / change the property value **let obj =** **{** **get propName() {** **// getter, the code executed on getting obj.propName** **},** **set propName(value) {** **// setter, the code executed on setting obj.propName = value** **}** **};** 3. An object property is a name, a value and a set of attributes. The value may be replaced by one or two methods, known as setter and a getter. **Example of getter** const student {	**4M** ***Explanation of setter with example 2M*** ***Explanation of getter with example 2M***

		firstname: 'abc',
		get getname()
		{
		return this.firstname;
		}
		};
		
		Example of setter
		
		const student
		{
		firstname: 'abc',
		set changename(nm)
		{
		this.firstname=nm;
		}
		};	

	c)
	
	Ans.
		Write a JavaScript program to check whether a number is positive, negative or zero using switch case.
		Note: Any other relevant logic shall be considered.
		<html>
		<body>
		<script type="text/javascript">
		var num=prompt("Enter number");
		switch (Math.sign(num))
		{
		case 1:
		alert("The number is Positive");
		break;
		case -1:
		alert("The number is Negative");
		break;
		default:
		alert("The number is Zero");
		}
		</script>
		</body>
		</html>
			4M
			
			Correct logic 2M
			
			Correct syntax 2M

SUMMER – 2024 EXAMINATION
MODEL ANSWER

Subject: Client Side Scripting Language **Subject Code:** **22519**

	d) **Ans.**	**State the use of following methods:** i) **charCodeAt( )** ii) **fromCharCode ( )** 1. **charCodeAt( )**: This method is used to return a unicode of specified character. **Syntax**: var code=letter.charCodeAt( ); **Example**: var ch=’a’; document.write(ch.charCodeAt( )); **Output**: 97 2. **fromCharCode( )**: This method is used to return a character for specified code. **Syntax**: var character=String.fromCharCode(code); **Example**: var character=String.fromCharCode(97); Document.write(ch); **Output**: a	**4M** ***Use of each method 2M***

3.	**a)**
	Ans.
		Attempt any <u>THREE</u> of the following:
		Explain Associative arrays in detail.
		Associative arrays are basically objects in JavaScript where indexes are replaced by user-defined keys.
		Syntax: var arr = {key1:'value1', key2:'value2'}
		Here, arr, is an associative array with key1, key2 being its keys or string indexes and value1 & value 2 are its elements.
		Example: var arr = { "Company Name": ‘Flexiple’, "ID": 123};
		
		The content or values of associative arrays is accessed by keys. An associative array is an array with string keys rather than numeric keys.
		For example:
		var arrAssociative = {
		"Company Name": 'Flexiple',
		"ID": 123
		};
			12
			4M
			
			Correct explanation 4M

		var arrNormal = ["Flexiple", 123]; Here, the keys of the associative array are “Company Name” & “ID” whereas in the normal array. The keys or index is 0 & 1.	

	b)
	
	Ans.
		Write a JavaScript function that checks whether a passed string is palindrome or not.
		Note: Any other relevant logic shall be considered
		function isPalindrome(str) {
		str = str.replace(/[^A-Za-z0-9]/g, '').toLowerCase();
		return str === str.split('').reverse().join('');
		}
		console.log(isPalindrome("A man, a plan, a canal, Panama")); // Output: true
		console.log(isPalindrome("racecar")); // Output: true
		console.log(isPalindrome("hello")); // Output: false
			4M
			
			Correct logic 2M
			
			Correct syntax 2M

	c) **Ans.**	**Explain how to add and sort elements in array with suitable example.** **Adding Elements to an Array:** In JavaScript, you can add elements to an array using various methods, such as push(), unshift(), or direct assignment to a specific index. **Using push():** The push() method adds one or more elements to the end of an array and returns the new length of the array. **Using unshift():** The unshift() method adds one or more elements to the beginning of an array and returns the new length of the array. **Using splice():** This method can be used to add new items to an array, and removes elements from an array. **Syntax:** arr.splice(start_index,removed_elements,list_of_elemnts_to_be_added);	**4M** ***Explanation of adding elements with suitable example*** ***Any one method 2M***

		Parameter:
		•The first parameter defines the position where new elements should be added (spliced in).
		•The second parameter defines how many elements should be removed.
		•The list_of_elemnts_to_be_added parameter define the new elements to be added(optional).
		
		Using length property:
		The length property provides an easy way to append a new element to an array.
		Example
		<script>
		var fruits = ["Banana", "Orange", "Apple", "Mango"];
		document.write(fruits+"
		");
		fruits[fruits.length] = "Kiwi";
		document.write(fruits+"
		");
		fruits[fruits.length] = "Chikoo";
		document.write(fruits);
		</script>
		Sorting Elements in an Array:
		JavaScript provides the sort() method to sort the elements of an array in place and returns the sorted array.
		Example
		let numbers = [5, 2, 8, 1, 4];
		numbers.push(7);
		console.log("Array before sorting:", numbers);
		numbers.sort((a, b) => a - b);
		console.log("Array after sorting:", numbers);
			Explanation of sorting elements with suitable example 2M

	d) Ans.	**Explain the term browser location and history in details.** **Window Location Object:** In JavaScript, the window.location object represents the current URL of the browser window. It provides properties and methods to manipulate the URL.	**4M**

		window.location.href:
		Returns the complete URL of the current page.
		window.location.hostname:
		Returns the domain name of the web server.
		window.location.pathname:
		Returns the path and filename of the current page.
		window.location.protocol:
		Returns the protocol (HTTP, HTTPS, etc.) of the current page.
		window.location.assign(url):
		Loads the specified URL.
		window.location.reload(forceReload):
		Reloads the current page.
		// Example: Changing browser location
		window.location.href = "https://example.com/page2";
		Window History
		The window.history object can be written without the window prefix.
		To protect the privacy of the users, there are limitations to how JavaScript can access this object.
		Some methods:
		history.back() - same as clicking back in the browser
		history.forward() - same as clicking forward in the browser
		Window History Back
		The history.back() method loads the previous URL in the history list.
		This is the same as clicking the Back button in the browser.
		Example:
		<html>
		<head>
		<script>
		function goBack() {
		window.history.back()
		}
			Explanation of location 2M
			
			Explanation of history 2M

```
</script>
</head>
<body>
<input type="button" value="Back" onclick="goBack()">
</body>
</html>
```

4. **a)** **Attempt any THREE of the following:** **12**

State what is frame? Explain how it can be created with suitable example. **4M**

Ans.

Note: Explanation of either <frameset> , <frame> or <iframe> shall be considered.

Definition 1M

Explanation with example 3M

A frame refers to an HTML element that allows the display of another HTML document within the current web page. Frames are implemented using the <frameset>, <frame> and <iframe> (Inline Frame) element in HTML.

Frames are used to divide browser window into multiple sections where each section is treated as window that can have independent contents. A frame can load separate HTML document in each frame in a window.

<frameset> tag : A frameset is defined as a set of frames inserted in an HTML web page. These frames can be in the form of rows and columns in different size. Frameset tells the browser how to divide the screen into multiple areas.
<frameset> ... </frameset>

Attributes:

cols=”pixels/percentage”	Specify number and size of columns in a frameset. Default value is 100% (1 column).
rows=”pixels/percentage”	Specify number and size of rows in a frameset. Default value is 100% (1 row).

<frame> tag : Frame tag is used to insert web page content in a frame. It is an empty tag.

Attributes:

src="URL"	Specify address of a web page to be displayed in a frame.
name=" string"	Specify name of the frame which can be used as target to open a link.

Code:

```
<html>
<frameset cols="25%,75%" >
<frame src="page1.html" name="f1">
<frame src="page2.html" name="f2">
</frameset>
</html>
```

OR

Creating Frames with <iframe> in HTML:

To create a frame using the <iframe> element, you specify the URL of the document you want to embed as the value of the src attribute.

Cross-Origin Restrictions: When embedding content from external sources, cross-origin restrictions may apply. This means that the embedded content must be served with appropriate CORS (Cross-Origin Resource Sharing) headers to allow it to be displayed within the frame.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Frame Example</title>
<style>
    /* Style the iframe */
iframe {
```

		width: 100%; height: 300px; /* Set the height as desired */ border: 1px solid #ccc; /* Add a border for clarity */ } </style> </head> <body> <!-- Create a frame with an embedded document --> <iframesrc="https://www.example.com"></iframe> </body> </html>	
	b) Ans.	**Explain the steps to create floating menu and chain select menu** A floating menu is a menu that remains visible as the user scrolls down a web page. It's often used for navigation or providing quick access to important content. Here's how you can create one: HTML Structure: Create the HTML structure for the menu. This typically involves using <nav> or elements for the menu container, and and elements for the menu items. CSS Styling: Use CSS to style the menu and make it float on the page. You can use position: fixed to fix the menu in place and top, bottom, left, or right properties to position it relative to the viewport. JavaScript (Optional): You can enhance the functionality of the floating menu with JavaScript. For example, you can add smooth scrolling to anchor links within the menu, or you can add animations to make the menu appear or disappear dynamically. **Example:** <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Floating Menu Example</title> <style>	**4M** ***Correct steps of each 2M***

```
        /* CSS styles for the floating menu */

.floating-menu {
            position: fixed;
            top: 0;
            left: 0;
            background-color: #333;
            padding: 10px;
            width: 100%;
            z-index: 1000; /* Ensure it's above other content */
        }
.menu-item {
            display: inline-block;
            margin-right: 10px;
color: #fff;
            text-decoration: none;
        }
</style>
</head>
<body>
<nav class="floating-menu">
<ul>
<li class="menu-item"><a href="#section1">Section 1</a></li>
<li class="menu-item"><a href="#section2">Section 2</a></li>
<!-- Add more menu items as needed -->
</ul>
</nav>
<section id="section1">
<h2>Section 1</h2>
<p>This is the content of section 1.</p>
</section>
<section id="section2">
<h2>Section 2</h2>
<p>This is the content of section 2.</p>
```

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

Chained Select Menu:

A chained select menu, also known as a dependent or cascading select menu, consists of multiple dropdown menus where the options in one dropdown menu depend on the selection made in another dropdown menu.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chained Select Menu Example</title>
</head>
<body>
<label for="country">Country:</label>
<select id="country">
<option value="">Select a country</option>
<option value="usa">USA</option>
<option value="uk">UK</option>
</select>

<label for="city">City:</label>
<select id="city" disabled>
<option value="">Select a city</option>
</select>
<script>

 // JavaScript code to handle the chained select menu
constcountrySelect = document.getElementById('country');
```

```
constcitySelect = document.getElementById('city');

// Data source (for demonstration purposes)
constcitiesByCountry = {
        'usa': ['New York', 'Los Angeles', 'Chicago'],
        'uk': ['London', 'Manchester', 'Birmingham']
    };

countrySelect.addEventListener('change', function() {
constselectedCountry = this.value;
        if (selectedCountry) {
citySelect.innerHTML = '';
const cities = citiesByCountry[selectedCountry];
            if (cities) {
citySelect.disabled = false;
citySelect.innerHTML += '<option value="">Select a city</option>';
cities.forEach(city => {
citySelect.innerHTML += `<option
value="${city}">${city}</option>`;
                });
            } else {
citySelect.disabled = true;
            }
        } else {
citySelect.disabled = true;
citySelect.innerHTML = '<option value="">Select a city</option>';
        }
    });
</script>
</body>
</html>
```

c) Ans. **Explain how to use banners for displaying advertisement.** **4M**

Following are the steps to insert banner advertisement in webpage.

		1) Create banner advertisement using a graphics tool such as PhototShop, Paint, etc. 2) Create an <img> element in web page with height and width to display banner advertisement. 3) Build JavaScript that loads and display banner advertisements. <html> <head> <title>Banner Advertisements</title> </head> <body bgcolor="#EEEEEE"> <imgsrc="ad.jpg"/> </body> </html>	***Correct explanation 4M***

	d)
	
	Ans.
		Write a JavaScript function to check whether a given address is a valid IP address or not.
		Note: Any other relevant logic shall be considered
		function isValidIPAddress(address) {
		const ipv4Regex = /^(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})$/;
		const match = address.match(ipv4Regex);
		if (match) {
		for (let i = 1; i<= 4; i++) {
		const part = parseInt(match[i]);
		if (part < 0 || part > 255 || isNaN(part)) {
		return false; // Invalid part
		}
		}
		return true;
		} else {
		return false;
		}
		}
			4M
			
			Correct logic 2M
			
			Correct syntax 2M

	e) Ans.	**Explain process to create status bar in JavaScript.** **Status Bar:** The status bar is located at the bottom of the browser window and is used to display a short message to visitors on a web page. Developers who are clever to utilize the status bar employ various techniques to incorporate the status bar in the design of their web page. Some developers display a message on the status bar when the web page first opens. Other developers might change the message to reflect whatever the visitor is doing on the web page. For example, if a user is filling registration form then status bar will display a text as 'User is on form filling section'. **Building a Static Message:** A static message appears when the web page opens and remains on the status bar until the web page is closed. The content of the status bar is the value of the window object's status property. To display a message on the status bar, assign the message to the status property of the window object. **Example:-** window.status= 'You are on home page'; <html> <head> <script type="text/javascript"> window.status='Welcome to Home Page'; </script> </head> <body> Hello welcome to JavaScript </body> </html>	**4M** ***Correct explanation 4M***
5.	**a)**	**Attempt any <u>TWO</u> of the following:** **Write HTML script that displays textboxes for accepting username and password. Write proper JavaScript such that when the user clicks on submit button**	**12** **6M**

	Ans.	**i) All textboxes must get disabled and change the color to 'RED; and with respective labels**
		ii) Prompt the error message if the password is less than six characters
		Note: Any other relevant logic shall be considered
		<html>
		<head>
		<script>
		function disableTxt()
		{
		document.getElementById("un").disabled = true;
		document.getElementById('un').style.color = "red";
		
		document.getElementById('aaa').style.color = "red";
		document.getElementById("pass").disabled = true;
		document.getElementById('pass').style.color = "red";
		document.getElementById('bbb').style.color = "red";
		}
		function validateform(){
		var username=document.myform.username.value;
		var password=document.myform.password.value;
		if (username==null || username==""){
		alert("Name can't be blank");
		return false;
		}else if(password.length<6){
		alert("Password must be at least 6 characters long.");
		return false;
		}
		}
		</script>
		</head>
		<body>
		<form name="myform" method="post" action="" onsubmit="return validateform()" >
		<label id = "aaa">Username:</label>
			Create textboxes 2M
			
			Disable textboxes with red color 2M
			
			Password validation 2M

		<input type="text" id="un" name="username"/> <label id = "bbb"> Password: </label> <input type="password" id="pass" name="password"/> <button onclick="disableTxt()">Disable Text field</button> </form> </body> </html>	

	b)
	
	Ans.
		Write a webpage that displays a form that contains an input for students rollno and names user is prompted to enter the input student rollno and name and rollno becomes value of the cookie.
		Note: Any other relevant logic shall be considered
		<html>
		<head><script>
		function writeCookie()
		{
		var d=new Date();
		d.setTime(d.getTime()+(1000*60*60*24));
		with(document.myform)
		{
		document.cookie="Roll No=" + student.value + ";expires="
		+d.toGMTString();
		} }
		function readCookie()
		{
		if(document.cookie=="")
		document.write("cookies not found");
		else
		document.write(document.cookie);
		}
		</script>
		</head>
			6M
			
			Create input text boxes for username and password 2M
			
			Set cookie 2M
			
			display 2M

		<body> <form name="myform" action=""> Enter your name: <input type="text" name="student"> Enter your Roll No: <input type="roll no" name="student"> <input type="Reset" value="Set C" type="button" onclick="writeCookie()"> <input type="Reset" value="Get C" type="button" onclick="readCookie()"> </form></body> </html>	
	c) **Ans.**	**Write a JavaScript to create rollover effect that involves text and images. When the user places his or her mouse pointer over a book title, the corresponding book images appears** ***Note: Any other relevant logic shall be considered*** <html> <head> <title> rollovers</title> </head> <body> <table border="1" width="100%"> <tbody> <tr valign="top"> <td width="50%"> <a><img height="500" src="motivation.png" width="900" name="clr"></td> <td> <u>Motivational book</u> <u>Educational book</u> </td>	**6M** ***Correct logic 3M*** ***Correct syntax 3M***

		</tr> </tbody> </table> </body> </html>	

6.	**a)**
	
	Ans.
		Attempt any <u>TWO</u> of the following:
		Explain following form control / elements with example Button, Text, TextArea, Select, Checkbox, Form.
		Note: Combined example including all controls / elements shall be considered.
		Button is created by using following code:
		<form method = "GET" action = ""><input type = "button" name = "MyButton" value = "Click" onclick = "msg()"><form>
		There are several types of button, which are specified by the type attribute:
		1. Button which corresponds to the graphic component.
		2. Submit, which is associated to the form and which starts the loading of the file assigned to the action attribute.
		3. Image button in which an image loaded from a file.
		A Button object also represents an HTML <button> element which is specified as follows:
		<button name = "btn" value = "MyButton" onclick = "msg()">
		Example:
		<html>
		<body>
		 Show a Push Button
		 The button below activates a JavaScript when it is clicked.
		<form>
		<input type="button" value="Click me" onclick="msg()">
		</form>
		<script>
		function msg()
		{
		alert("Hello world!");
		}
		</script>
			12
			6M
			
			Explanation of each control with example 1M

</body>
</html>

Text:
Input "text" is an object to enter a single line of text whose content will be part of form data.
In html a text is created by following code:
<input type="text" name="textname" id="textid" value=" assign_value" />

Example:

```
<script type="text/javascript">
function changeText()
{
var userInput = document.getElementById('userInput').value;
document.getElementById('vp').innerHTML = userInput;
}
</script>
<input type='text' id='userInput' value='Enter Text Here' />
<p>Welcome <b id='vp'>JavaScript</b></p>
<input type='button' onclick='changeText()' value='Change Text'/>
</script>
```

TextArea:
The Textarea object represents an HTML <textarea> element.
The <textarea> tag indicates a form field where the user can enter a large amount of text.
You can access a <textarea> element by using getElementById()

Example:

```
<html>
<body>
<textarea cols="30" rows="5" wrap="hard" readonly="yes" disabled="yes">
As you can see many times word wrapping is often the desired look for your textareas. Since it makes everything nice and easy to read and preserves line breaks.
```

</textarea>
</body>
</html>

Checkbox:
<input> elements of type checkbox are rendered by default as boxes that are checked (ticked) when activated. A checkbox allows you to select single values for submission in a form (or not).
Syntax for creating checkbox is:
<input type="checkbox" id="myCheck" onclick="myFunction()">
A checkbox can have only two states:
1. Checked
2. Unchecked

Example:

```
<html>
<body>
<div>
<br>
<input type="checkbox" name="program" id="it" value="IT">
<label for="it">Information Tech</label><br>
<input type="checkbox" name="program" id="co" value="CO" checked>
<label for="co">Computer Engg</label><br>
<input type="checkbox" name="program" id="ej" value="EJ">
<label for="ej">Electronics</label><br>
<button onclick="validate();">Validate</button>
</div>
<div id="status">
</div>
<script>
function validate()
{
var elements = document.getElementsByName("program");
var statusText = " ";
```

```
for (var index=0;index <elements.length;index++)
{
statusText = statusText +
elements[index].value+"="+elements[index].checked+"<br>";
}
document.getElementById("status").innerHTML = statusText;
}
</script>
</body>
</html>
```

Select:

Form SELECT elements (<select>) within your form can be accessed and manipulated in JavaScript via the corresponding Select object.

To access a SELECT element in JavaScript, use the syntax:

document.myform.selectname //where myform and selectname are names of your form/element.

document.myform.elements[i] //where i is the position of the select element within form

document.getElementById("selectid") //where "selectid" is the ID of the SELECT element on the page.

Example:

```
<html>
<body>
<select id="programs" size="5">
<option>Computer Engineering</option>
<option>Information Technology</option>
<option>Chemical Engineering</option>
<option>Electronics &TeleComm.</option>
</select>
<p>Click the button to disable the third option (index 2) in the dropdown list.</p>
<button onclick="myFunction()">Disable Option</button>
<script>
```

		function myFunction()
		{
		var x = document.getElementById("programs").options[2].disabled = true;
		document.getElementById("programs").options[2].style.color = "red";
		}
		</script>
		</body>
		</html>
		
		Form:
		A form is a section of an HTML document that contains elements such as radio buttons, text boxes and option lists. HTML form elements are also known as controls.
		Elements are used as an efficient way for a user to enter information into a form. Typical form control objects also called "widgets" includes the following:
		• Text box for entering a line of text.
		• Push button for selecting an action.
		• Radio buttons for making one selection among a group of options.
		• Check boxes for selecting or deselecting a single, independent option.
		The <form> element can contain one or more of the following form elements:
		· <input> · <textarea> · <button> · <select> · <option> · <fieldset> · <label>
		· <legend>
		Syntax:
		<form name = "myform" id = "myform" action = "page.html" onSubmit = "test()"> -----objects---- </form>	

	b)	**Write a JavaScript for protecting web page by implementing the following steps:** **i) Hiding your source code**	**6M**

	Ans.	**ii) Disabling the right MouseButton** iii) **Hiding JavaScript** **i) Hiding your source code:** • Every developer has to admit that, on occasion, they've peeked at the code of a web page or two by right-clicking and choosing View Source from the context menu. • In fact, this technique is a very common way for developers to learn new techniques for writing HTML and Javascript. However, some developers don't appreciate a colleague snooping around their code and then borrowing their work without permission. This is particularly true about javascript, which are typically more time-consuming to develop than using HTML to build a web page. • In reality, you cannot hide your HTML code and JavaScript from prying eyes, because a clever developer can easily write a program that pretends to be a browser and calls your web page from your web server, saving the web page to disk, where it can then be opened using an editor. Furthermore, the source code for your web page—including your JavaScript—is stored in the cache, the part of computer memory where the browser stores web pages that were requested by the visitor. • A sophisticated visitor can access the cache and thereby gain access to the web page source code. • However, you can place obstacles in the way of a potential peeker. First, you can disable use of the right mouse button on your site so the visitor can't access the View Source menu option on the context menu. This hide both your HTML code and your JavaScript from the visitor. Nevertheless, the visitor can still use the View menu's Source option to display your source code. In addition, you can store your JavaScript on your web server instead of building it into your web page. The browser calls the JavaScript from the web server when it is needed by your web page. • Using this method, the JavaScript isn't visible to the visitor, even if the visitor views the source code for the web page. **ii) Disabling the right Mouse Button:** The following example shows how to disable the visitor's right mouse button while the browser displays your web page. All the	***Hiding source code 2M*** ***Disabling Mouse button 2M*** ***Hiding JavaScript 2M***

action occurs in the JavaScript that is defined in the <head> tag of the web page.

Example:

```
<html>
<head>
<script>
window.onload = function()
{
document.addEventListener("contextmenu", function(e)
{
e.preventDefault();
}, false);}
</script>
<body>
<h3>Right click on screen,Context Menu is disabled</h3>
</body>
</html>
```

The preventDefault() method cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur.

iii) Hiding JavaScript:

You can hide your JavaScript from a visitor by storing it in an external file on your web server. The external file should have the .js file extension. The browser then calls the external file whenever the browser encounters a JavaScript element in the web page. If you look at the source code for the web page, you'll see reference to the external .js file, but you won't see the source code for the JavaScript.

The next example shows how to create and use an external JavaScript file. First you must tell the browser that the content of the JavaScript is located in an external file on the web server rather than built into the web page. You do this by assigning the file name that contains the JavaScript to the src attribute of the <script> tag.

		Next, you need to define empty functions for each function that you define in the external JavaScript file.
		
		webpage.html
		<html>
		<head>
		<script src="mycode.js" languages="javascript" type = "text / javascript">
		</script>
		<body>
		  Right Click on screen, Context Menu is disabled
		</body>
		</html>
		mycode.js
		window.onload=function()
		{
		document.addEventListener("contextmenu", function(e)
		{
		e.preventDefault();
		}, false);
		}	

	c) **Ans.**	**Develop a JavaScript to create rotating Banner Ads with URL links.** ***Note: Any other correct logic / program shall be considered*** <html> <head> <title>Link Banner Ads</title> <script language="Javascript" type="text/javascript"> Banners = new Array('1.jpg','2.jpg') BannerLink = new Array('google.com/', 'msbte.org.in/'); CurrentBanner = 0; NumOfBanners = Banners.length; function LinkBanner() { document.location.href ="http://www." +	**6M** ***Correct program 6M***

		BannerLink[CurrentBanner];
		}
		function DisplayBanners() {
		if (document.images) {
		CurrentBanner++
		if (CurrentBanner == NumOfBanners) {
		CurrentBanner = 0
		}
		document.RotateBanner.src= Banners[CurrentBanner]
		setTimeout("DisplayBanners()",1000)
		} }
		</script>
		</head>
		<body onload="DisplayBanners()" >
		
		<imgsrc="1.jpg"
		width="400" height="75" name="RotateBanner" />
		
		</body></html>	

WINTER – 2023 EXAMINATION
Model Answer – Only for the Use of RAC Assessors

Subject Name: Client Side Scripting **Subject Code:** 22519

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.
9) For programming language papers, in answer session comments and output is not required.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**State meaning of each token of the following statement and describe it:** **i) ++a;** **ii) document. bgcolor;**	**2 M**
	Ans	(i) ++ is the increment operator and a is the operand, hence a++ will increment the value of a by 1. (ii) The bgColor property specifies the background color of HTML document.	Anyone explained - 1 M each.
	b)	**Write and explain syntax of prompt( ) method in JavaScript.**	**2 M**
	Ans	• **prompt()** method is used to display a dialogue box to the user to prompt them to an input. • It has two buttons "OK" and "CANCEL", if the user click on the "OK" button then it will return the inputed value , if the user clicks on the "CANCEL" button then it will return a null value • Syntax: prompt(text)	Definition- 1 M Syntax- 1 M

- Example:

```
<script>
 var name = prompt ("Enter a name");
 document. Write(name);
</script>
```

c) List various assignment operators supported by JavaScript, explain any two with the help of suitable example. — **2 M**

Ans — **assignment operators supported by JavaScript** — List – 1 M; Explain any 2 - 1 M

Operators	Meaning
=	Equal to
+=	Plus Equal to
-=	Minus Equal to
/=	Divide Equal to
%=	Modulus Equal to
*=	Multiply Equal to

Example:

```
<script>
// = is used to assign a value to a variable
 var num = 10;
document.write(num); // num 10
/* += first add the value to the existing value of the variable then assign it the new added value */
 num +=10; document.write(num); // num 20
</script>
```

d) Differentiate between shift() and push() methods of an Array object. — **2 M**

Ans — Any two points- 1 M each

shift	push
1. shift method is used to delete the first element of an existing array	1. push method is used to insert zero or more element at the end of an existing array
2. Syntax: arr.shift();	2. Syntax: arr.push(element1,element2,…,elementn);

3. Example:
<script>
var arr = [1,2,3,4];
document.write(arr);
arr.shift();
document.write(arr);
</script>
	3. Example:
	<script>
	var arr = [1,2,3,4];
	document.write(arr);
	arr.push(5,6);
	document.write(arr);
	</script>

	e)	**State the meaning of "Defining a function". Explain with the help of an example.**	**2 M**

	Ans	• A function is a block of code that takes some input to perform some certain computation.
		• The main purpose of the function is to put commonly used or repeatedly used task in a function, so instead of writing the code again and again we can call it instead.
		• The function can be defined as followed:
		Syntax:
		function func_name(parameter1 ,parameter2,…,parametern)
		{
		//code
		}
		Example:
		<script>
		function add(num1,num2)
		{
		return num1 + num2;
		}
		add(1,2);
		</script>
			Explanation- 1 M
			Example- 1 M

	f)	**Give syntax of and explain the use of small "with" clause.**	**2 M**

	Ans	**"with"** clause is used to directly access the properties and method of an object.
		Syntax:
		with (object)
		{
		//object
		}
		Example:
		<script>
		var person ={ name:"Abc", age:18
		}
		with(person){ docment.write(name); docment.write(age);
		}
		</script>
			Explanation- 1 M
			Syntax- 1 M

	g)	**With the help of suitable example explain the Date object and any two methods of Date object.**	**2 M**
	Ans	**"Date"** object is used when we want to perform some operation on date, it has various method which helps you performs some task related to date, Example:	Explanation- 1 M Example- 1

		<script>
		var date = new Date();
		document.write(date.getDate()); // get the current date
		document.write(date.getFullYear()); // get the current Year
		document.write(date.getMinutes()); // get the current minutes
		</script>
			M

2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write a JavaScript that accepts a number and displays addition of digits of that number in a message box.**	**4 M**

	Ans	<html>
		<body>
		<script>
		var num = prompt("Enter a 2 digit number:"); //accept number from user
		var digits = num.split();
		var firstdigit = Integer(digits[0]);
		var seconddigit = Integer(digits[1]);
		var addition = firstdigit+seconddigit;
		alert("The addition is "+addition); //display result in message box
		</script>
		</body>
		</html>
			Relevant code- 4 M

	b)	**Describe the navigator object in JavaScript. Describe the methods of navigator object which is used to display browser name and version.**	**4 M**
	Ans	• Navigator object is a representation of user's browser. • Navigator is an object of window. ➢ **Methods of navigator object- Only one method is supported by Navigator Object.** i) javaEnabled()- returns true if the browser has java enabled. • We use appName to display browser name. appName is a property of navigator. • appName property returns browser name. • We use appVersion to display browser version. appVersion is a property of navigator. • appVersion property returns browser version. • **Example:-** <html> <body> <script> document.write("Browser name: "+navigator.appName); document.write("Browser Version: "+navigator.appVersion); </script>	Explanation - 2 M Method – 2 M

		</body> </html>	
	c)	**Give syntax of and explain for-in loop in javascript with the help of suitable example.**	**4 M**

	Ans	**For-in Loop:**
		Syntax:-
		For(x in object){
		//code to be executed
		}
		• For-in loop is used to loop through the properties of an object.
		• In syntax, variable represents the property name, and object is the object being iterated.
		• It's handy for tasks like accessing or manipulating object properties.
		Example:-
		<html>
		<body>
		<script>
		Var car = { Brand: 'Toyota', Model: 'Camry', Year: 2022
		};
		For(key in car){ Document.write('${key}: ${car[key]}');
		}
		</script>
		</body>
		</html>
			Explanation - 2 M
			Syntax – 1 M
			Example-1 M

	d)	**Write an HTML script that accepts Amount, Rate of interest and Period from user. When user submits the information a JavaScript function must calculate and display simple interest in a message box. (Use formula S.I. = PNR/100)**	**4 M**

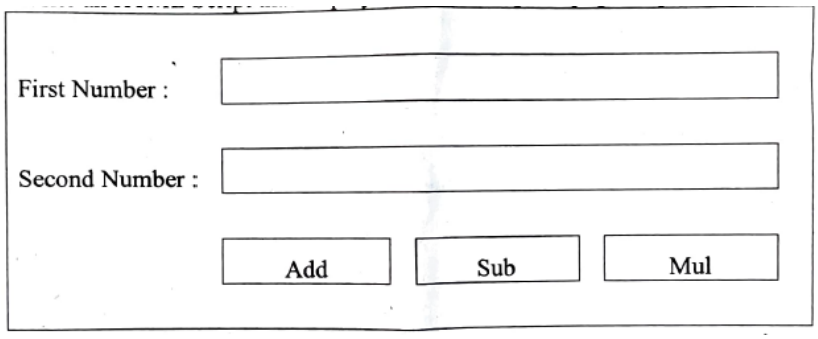
	Ans	<html>
		<body>
		<script>
		var P = parseInt(prompt("Enter the principal amount:"));
		var N = parseInt(prompt("Enter the period:"));
		var R = parseInt(prompt("Enter the Rate of interest:"));
		var SI =(P*N*R)/100;
		alert("Simple Interest is "+SI);
			Relevant code- 4 M

```
</script>

</body>

</html>
```

OR

```
<html>
  <head>
    <script>

        function interest()
       {
        var P, N, R;
        P= parseInt(document.getElementById("pr").value);
        N = parseInt(document.getElementById("period").value);
        R = parseInt(document.getElementById("ri").value);
       var SI =(P*N*R)/100;
        alert("Simple Interest is="+SI);
       }
       </script>
  </head>
  <body>
    <p>Principal Amount:<input id="pr"></p>
    <p>Period in Year: <input id="period"></p>
    <p>Rate of Interst: <input id="ri"></p>
    <button onclick="interest()"> Simple Interest</button>
  </body>
</html>
```

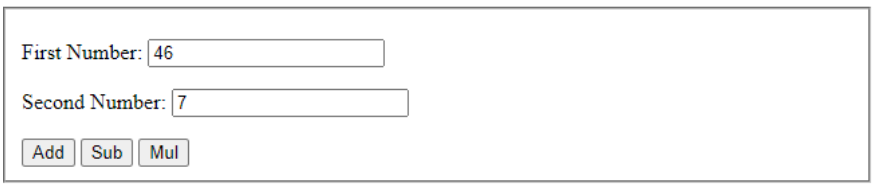
Output:

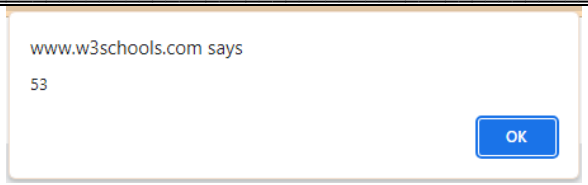
3.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write an HTML Script that displays the following webpage output:**  **The user enters two numbers in respective text boxes. Write a JavaScript such that when user clicks "add", a message box displays sum of two entered numbers, if the user clicks on "sub". Message box displays subtraction of two numbers and on clicking "mul" the message box displays multiplication of two numbers.**	**Designing and calling function in onClick() - 1M** **Writing function for add,sub,mul- 03M**

	Ans	<html>
		<head>
		<script>
		function add()
		{
		var num1, num2, r;
		//to accept 2 values and stored in variable num1 and num2
		num1 = parseInt(document.getElementById("firstnumber").value);
		num2 = parseInt(document.getElementById("secondnumber").value);
		r= num1 + num2;
		alert(r);
		}
		function sub()
		{
		var num1, num2, r;
		num1 = parseInt(document.getElementById("firstnumber").value);
		num2 = parseInt(document.getElementById("secondnumber").value);	

```
        r = num1 - num2;
        alert(r);
      }
        function mul()
      {
        var num1, num2, r;
        num1 = parseInt(document.getElementById("firstnumber").value);
        num2 = parseInt(document.getElementById("secondnumber").value);
        r = num1 * num2;
        alert(r);
      }
      </script>
  </head>
  <body>
  <fieldset>
    <p>First Number: <input id="firstnumber"></p>
    <p>Second Number: <input id="secondnumber"></p>
  //onClick() event to perform addition, subtraction and multiplication
    <button onclick="add()">Add</button>
    <button onclick="sub()">Sub</button>
    <button onclick="mul()">Mul</button>
   </fieldset>
  </body>
</html>
```

Output:





b) **Write a JavaScript that accents user's first name and domain name of Organization from user., The JavaScript then forms email address as <firstname@domain> name and displays the results in the browser window.** **4 M**

Ans

```
<html>
  <head>
    <script>
  //Email Address Function
function myemail()
{
  var s1 = document.getElementById("first").value;
  var s2 = document.getElementById("domain").value;
  var s3=s1+"@"+s2;
  document.getElementById("email").innerHTML = s3;
}
   </script>
  </head>
  <body>
     <p>First Name: <input id="first"></p>
    <p>Domain Name: <input id="domain"></p>
    <button onclick="myemail()">create email id</button>
    <br>  Email ID is
    <p id="email"> </p>
 </body>
</html>
```

Accept values-1M

Formation of email id-2M

Calling function in onClick()-1M

Output:

First Name: mahak

Domain Name: zita.edu.in

create email id

Email ID is

mahak@zita.edu.in

c) Differentiate between substring() and substr() method of a string class. Give Suitable example of each. **4 M**

Ans

Parameters	substring()	substr()
Use	It is used to extract the specified substring within a string	It is used to extract a part of the string
Syntax	str.substring(start, end);	str.substr(start, len);
Parameters	Its parameters are the start and end position of the substring that we want to extract.	It takes parameters as starting index of the part which we want to extract and the length till which we want to extract the string part.
index	This method cannot handle negative indexes.	This method can handle negative indexes.

3 points-3M

Example-1M

Example:

```
<script>
var a="Javascript";
document.write("Using substring()="+a.substring(2,6));
document.write("<br>Using substr()="+a.substr(2,6));
</script>
```

Output:

Using substring()=vasc
Using substr()=vascri

d) State what is a cookie? Explain its need. State characteristics of persistent cookies. **4 M**

	Ans	**Cookie:** **Cookies** are small text files that websites use to store information on a user's computer or device. Cookies can be used for a variety of purposes, such as remembering a user's preferences, tracking their behavior on a website, and enabling personalized advertising. There are two main types of cookies: session cookies and persistent cookies. **Need of Cookie:** Cookies are built specifically for web browsers to track, personalize and save information about each user's session. A "session" is the word used to define the amount of time you spend on a site. Cookies are created to identify you when you visit a new website. **Characteristics of persistent cookies:** • Persistent cookies can be used to store information that needs to be accessed across multiple browsing sessions, such as login credentials and language preferences. • Persistent cookies can be used to track a user's behavior on a website over time, which can be used to personalize the user's experience. • Persistent cookies can be used to remember a user's preferences, such as their preferred layout or font size.	Definition-1M Need-1M Charactristics-2M
4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write a JavaScript that accepts a string and searches for the pattern "MSBTE" in the given string using regular expressions. If the pattern is found, JavaScript will display that "Pattern is found" else display "Pattern is not found".**	**4 M**

	Ans	<html>
		<head>
		<title>JavaScript RegExp exec Method</title>
		</head>
		<body>
		<script type = "text/javascript">
		var str= prompt("Enter a string");
		var re = new RegExp("MSBTE", "g");
		var result = re.exec(str);
		if(result=="MSBTE")
		{
		document.write("pattern is found");
		}
		else
		{
		document.write("pattern is not found");
		}
			Accept string-1M
			
			Correct RegEx-2M
			
			Condition checking-1M

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

b) **List and state various properties of a window object. Write a JavaScript that opens a new popup window with message WELCOME To SCRIPTING" when the page loads and a new popup window displaying message "FUN WITH SCRIPTING when the page unloads.** **4 M**

Ans **Window Object Properties**

properties of a window object-1M

calling onload ()-1M

calling onunload()-1M

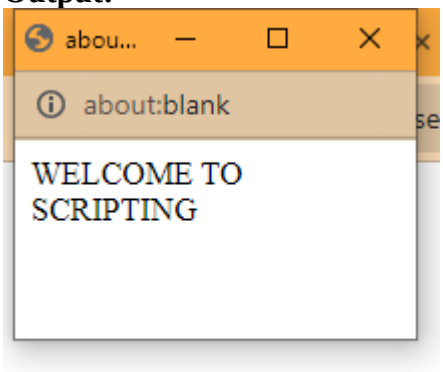
valid function definition-1M

Property	Description
Document	It returns the document object for the window (DOM).
Frames	It returns an array of all the frames including iframes in the current window.
Closed	It returns the Boolean value indicating whether a window has been closed or not.
History	It returns the history object for the window.
innerHeight	It sets or returns the inner height of a window's content area.
innerWidth	It sets or returns the inner width of a window's content area.
Length	It returns the number of frames in a window.
Location	It returns the location object for the window.
Name	It sets or returns the name of a window.
Navigator	It returns the navigator object for the window.
Opener	It returns a reference to the window that created the window.
outerHeight	It sets or returns the outer height of a window, including toolbars/scrollbars.
outerWidth	It sets or returns the outer width of a window, including toolbars/scrollbars.
Parent	It returns the parent window of the current window.
Screen	It returns the screen object for the window.
screenX	It returns the X coordinate of the window relative to the screen.
screenY	It returns the Y coordinate of the window relative to the screen.
Self	It returns the current window.
Status	It sets the text in the status bar of a window.
Top	It returns the topmost browser window that contains frames.

Code:

```
<html>
<body onload="openWin()" onunload = "myWindow.document.write("FUN WITH SCRIPTING");">

    <script>
    var myWindow;
    function openWin()
    {
      myWindow = window.open("", "myWindow", "width=200,height=100");
      myWindow.document.write("<p>WELCOME TO SCRIPTING</p>");
    }
    function closeWin()
    {
      myWindow.close();
   alert("Fun with SCripting");
    }
    </script>
    </body>
    </html>
```

Output:



c) Write an HTML script that displays names of different brands of Laptop and an image by default as : **4 M**

- Lenovo
- HP
- DELL

Image

When the mouse moves over the specific brand name the script must display the image of respective Laptop in the adjacent box.

Ans **Code:**

```
<html>
<head>
```

Designing of form-1M

Any relevant

		<title>
		text rollovers</title>
		<script>
		b=new Image;
		r=new Image;
		g=new Image;
		if(document.images)
		{
		b.src='vv.png';
		r.src='hp.png';
		g.src='dell.png';
		}
		else
		{
		b.src='';
		r.src='';
		g.src='';
		document.clr='';
		}
		
		</script>
		</head>
		<body>
		<table border="0" width="100%">
		<tbody>
		<tr valign="top">
		
		<td><H2>
		
		<u>Lenovo</u>
		
		<u>HP</u>
		
		<u>DELL</u>
		</H2>
		</td>
		<td width="50%">
		<a><img height="200" src="vv.png" width=400" name="clr"></td>
		</tr>
		</tbody>
		</table>
		</body>
		</html>
		
		Output:
		When mouse over to HP, hp laptop image will displayed.
			logic-3M

		• **Lenovo** • **HP** • **DELL**  • **Lenovo** • **HP** • **DELL** 	
	d)	**Give syntax of and explain the use of SetTimeOut() function with the help of suitable example.**	**4 M**

	Ans	**Use of setTimeout()**
		
		The setTimeout() method executes a block of code after the specified time. The method executes the code only once.
		
		The commonly used syntax of JavaScript setTimeout is:
		
		setTimeout(function, milliseconds);
		
		Its parameters are:
		• function - a function containing a block of code
		• milliseconds - the time after which the function is executed
		
		The setTimeout() method returns an intervalID, which is a positive integer.
		
		Example: Display a Text Once After 3 Second
		
		<script>
		function greet()
		{
		document.write('Hello world');
		}
		setTimeout(greet, 3000);
		document.write('This message is shown first');
		</script>
			Use of SetTimeOut()-1M
			
			Syntax-1M
			
			Example-2M

		In the above program, the setTimeout() method calls the greet() function after 3000 milliseconds (3 second). Hence, the program displays the text Hello world only once after 3 seconds.	
	e)	**State the use of hiding the JavaScript. Explain the steps needed to accomplish it and describe the process.**	**4 M**

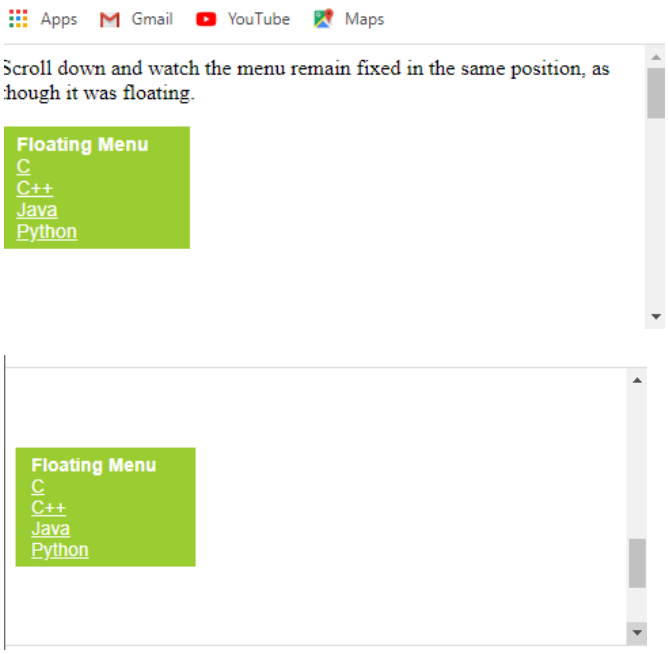
	Ans	You can hide your JavaScript from a visitor by storing it in an external file on your web server. The external file should have the .js file extension. The browser then calls the external file whenever the browser encounters a JavaScript element in the web page. If you look at the source code for the web page, you'll see reference to the external .js file, but you won't see the source code for the JavaScript.
		
		webpage.html
		<html>
		<head>
		<script src="mycode.js" languages="javascript" type="text/javascript">
		</script>
		<body>
		  Right Click on screen, Context Menu is disabled
		</body>
		</html>
		mycode.js
		window.onload=function()
		{
		document.addEventListener("contextmenu", function(e)
		{
		e.preventDefault();
		}, false);
		}
			Use-2M
			
			Steps with code-2M

5.		**Attempt any TWO of the following:**	**12 M**
	a)	**Write a JavaScript that demonstrates use of floating menu alongwith respective HTML script.**	**6 M**

	Ans	**Code:**
		<html>
		<title>Example</title>
		<style>
		body {
		background-image: url('/pix/samples/bg1.gif');
		}
		main {
		margin-bottom: 200%;
		}
		.floating-menu {
			Any relevant logic or example-6M

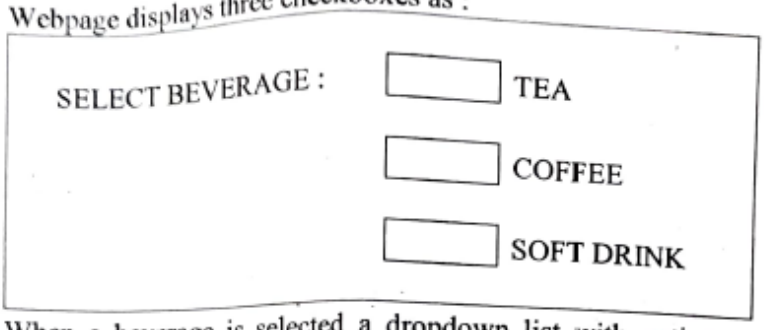
```
    font-family: sans-serif;
    background: yellowgreen;
    padding: 5px;;
    width: 130px;
    z-index: 100;
    position: fixed;
   }
   .floating-menu a,
   .floating-menu h3 {
    font-size: 0.9em;
    display: block;
    margin: 0 0.5em;
    color: white;
   }
</style>
<main>
  <p>Scroll down and watch the menu remain fixed in the same position, as though it was floating.</p>
  <nav class="floating-menu">
    <h3>Floating Menu</h3>
    <a href="c_sub.txt">C</a>
    <a href="C++_sub.txt">C++</a>
    <a href="java_sub.txt">Java</a>
    <a href="python_sub.txt">Python</a>
  </nav>
</main>
```

Output:



	b)	**Form regular expressions for following :**	**6 M**

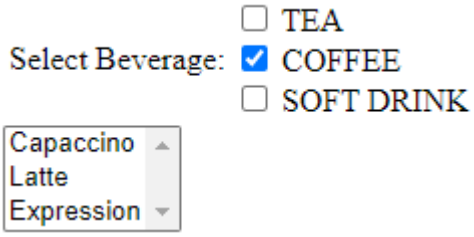
		i) Validation of email address. **ii) Validation of adhaar card. Format is** **dddd - dddd - dddd** **(iii) Validation of phone number. Format is** **(ddd) -(ddddddddd)**	
	Ans	**i) Regular expression for Validation of email address:** var validRegex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{\|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/; **ii) Validation of adhaar card Format is** **dddd - dddd – dddd** Regular expression for Validation of adhaar card /^[2-9]{1}[0-9]{3}\s[0-9]{4}\s[0-9]{4}$/gm **iii) Validation of phone number. Format is** **(ddd) -(ddddddddd)** Regular expression for Validation of phone number: var re = /^[\+]?[(]?[0-9]{3}[)]?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/im;s	Any valid regex for emailed-2M Any valid regex for adhaar card-2M Any valid regex for phone no-2M

	c)	(c) Write HTML and respective JavaScript such that (i) Webpage displays three checkboxes as :  (ii) When a beverage is selected a dropdown list with options as below appears on page : (a) If "TEA" option is checked dropdown list contains Green tea Milk tea Black tea (b) If "COFFEE" option is selected dropdown list contains. Capaccino Latte Expression (c) If "SOFT DRINK" option is selected dropdown list contains MAAZA SPRITE COCA-COLA	**6 M**

	Ans	Code:
		<html>
		<body>
		<html>
		<script type="text/javascript">
		function modifyList(x)
		{
		with(document.forms.myform)
		{
		if(x ==1)
		{
		optionList[0].text="Green Tea";
		optionList[0].value=1;
		optionList[1].text="Milk Tea ";
		optionList[1].value=2;
		optionList[2].text="Black Tea";
		optionList[2].value=3;
		}
		
		if(x ==2)
		{
		optionList[0].text="Capaccino";
		optionList[0].value=1;
			Designing of form-2M
			
			Any relevant script-4M

```
        optionList[1].text="Latte";
        optionList[1].value=2;
        optionList[2].text="Expression";
        optionList[2].value=3;
        }
  if(x ==3)
        {
        optionList[0].text="MAAZA";
        optionList[0].value=1;
        optionList[1].text="SPRITE";
        optionList[1].value=2;
        optionList[2].text="COKA-COLA";
        optionList[2].value=3;
        }

        }
        }

        </script>
        </head>
        <body>
        <form name="myform" action=" " method="post">
  <table><tr><td>
  Select Beverage: </td><td>
  <input type="checkbox" name="grp1" value=1 checked="true"
onclick="modifyList(this.value)"> TEA

 <input type="checkbox" name="grp1" value=2 onclick="modifyList(this.value)">
COFFEE
 <input type="checkbox" name="grp1" value=3 onclick="modifyList(this.value)">
SOFT DRINK

  </td></tr></table>
        <select name="optionList" size="3">
        <option value=1>Kiwi
        <option value=1>Pine-Apple
        <option value=1>Apple
  </tr> </table>
   </select>
        </form>
        </body>
        </html>
```

Output:



6.		**Attempt any TWO of the following:**	**12 M**
	a)	**List and explain any six form events.**	**6 M**
	Ans		One event with explanation-1M each

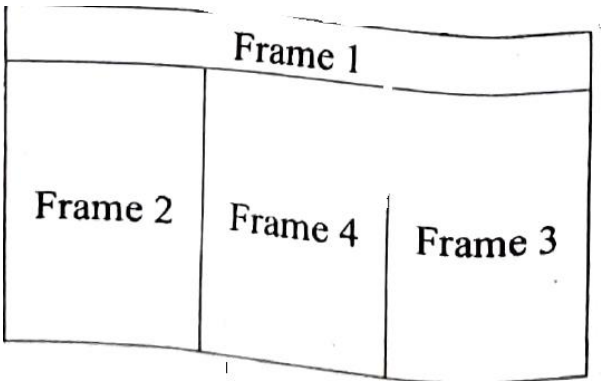
Event Handler	Description
onfocus	When the user focuses on an element
onsubmit	When the user submits the form
onblur	When the focus is away from a form element (The onblur event occurs when an object loses focus.)
onchange	When the user modifies or changes the value of a form element
keydown	The event occurs when the user is pressing a key
keypress	The event occurs when the user presses a key
keyup	The event occurs when the user releases a key
onclick	Fires on a mouse click on the element
ondblclick	Fires on a mouse double-click on the element
onmousedown	Fires when a mouse button is pressed down on an element
onmousemove	Fires when the mouse pointer is moving while it is over an element
onmouseout	Fires when the mouse pointer moves out of an element
onmouseover	Fires when the mouse pointer moves over an element
onmouseup	Fires when a mouse button is released over an element
onwheel	Fires when the mouse wheel rolls up or down over an element
oncontextmenu	oncontextmenu event occurs when the user right-clicks on an element to open the context menu.

	b)	**Write a JavaScript that sets a crawling status bar message to the webpage. Message is "Welcome to the Mystic World of JavaScript". The message must start crawling when the webpage gets loaded.**	**6 M**

	Ans	<html>
		<head>
		<title>Scrolling Text</title>
		<script language="JavaScript">
		var scrollPos = 0; // initial position to start status bar
		var maxScroll = 100; // maximum allowed position
			Any relevant script-6M

```
var blanks = "";
function scrollText(text, milliseconds)
{
window.setInterval("displayText('"+text+"')", milliseconds);
}
function displayText(text)
{
window.defaultStatus = blanks + text;
++scrollPos;
blanks += " ";
if(scrollPos > maxScroll)
{
scrollPos = 0;
blanks = "";
}
}
</script>
</head>
<body onload="scrollText('Welcome to the Mystic World of JavaScript'.', 300)">
<p>Watch the text scroll at the bottom of this window!</p>
</body>
</html>
```

c) **i)Design frameset tag for representing following layout:** **6 M**



ii) List any three properties of regular expression objects and state their use.

Ans **i)**

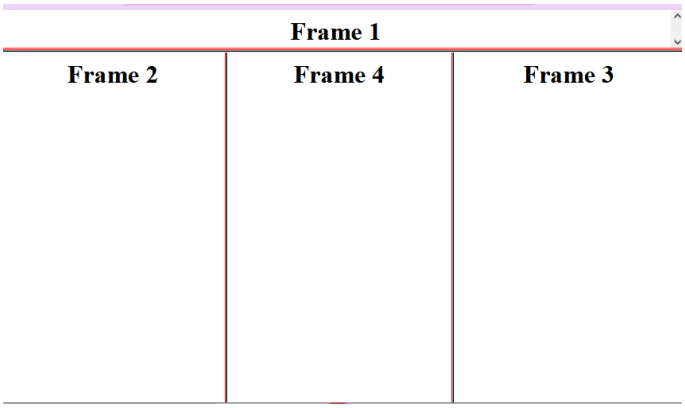
Code:

```
<html>
<head>
<title>Create a Frame</title>
</head>
<frameset rows="10%,*" border="5" frameborder="1" bordercolor="red" noresize>
<frame src="frame1.html">
<frameset cols="33%,33%,*" border="3">
<frame src="frame2.html">
<frame src="frame4.html">
```

i) correct html code-3M

ii) 3 properties with use-3M each

```
<frame src="frame3.html">
</frameset>
</frameset>
</html>
```



ii)

property	Property & Description
constructor	Specifies the function that creates an object's prototype.
global	Specifies if the "g" modifier is set.
ignoreCase	Specifies if the "i" modifier is set.
lastIndex	The index at which to start the next match.
multiline	Specifies if the "m" modifier is set.
source	The text of the pattern.

Example:

```
<html>
  <head>
    <title>JavaScript RegExp ignoreCase Property</title>
  </head>
    <body>
    <script type = "text/javascript">
      var re = new RegExp( "string" );

      if ( re.ignoreCase )
    {
```

```
            document.write("Test1-ignoreCase property is set");
          } else
         {
            document.write("Test1-ignoreCase property is not set");
          }

          re = new RegExp( "string", "i" );

          if ( re.ignoreCase )
        {
            document.write("<br/>Test2-ignoreCase property is set");
          } else
         {
            document.write("<br/>Test2-ignoreCase property is not set");
          }
        </script>
      </body>
    </html>
```