Table of Contents

2 Marks Questions

1.State the use of method in javascript with the help of suitable example.

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 ;
}
```

2. List & Explain datatypes in JavaScript.

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

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.

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.

3. Write a simple calculator program using switch case in JavaScript.

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

4.Write a program using sort method of array object.

```
<html>
<body>
 <script>
 var array =[5,1,9,7,5];
 // sorting the array
 sorted = array.sort();
 document.write(sorted);
 </script>
</body>
</html>
```

5. Describe property Getters & Setters.

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;
},
```

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

6. Enlist & explain the use of any two Intrinsic JavaScript Functions.

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

7. Describe browser location object.

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>
 <h1>The Window Location Object</h1>
 <p id="demo"></p>
<script>
 let origin = window.location.origin;
 document.getElementById("demo").innerHTML = origin;
</script>
</body>
</html>
```

8.List any four features of Java script

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

9. List the comparison operators in Java script.

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

10. Write a Java script to create person object with properties firstname, lastname, age, eyecolor, delete eyecolor property and display remaining properties of person object.

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>
```

11. Write a Java script that initializes an array called flowers with the names of 3 flowers. The script then displays array elements.

```
<html>
<head>
<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] + '<br>');
}
</script>
```

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

12. Write Javascript to call function from HTML.

```
<html>
<head>
<title>Calling function from HTML</title>
<script>
function welcome()
{
alert("Welcome students");
}
function goodbye()
{
alert("Bye");
}
</script>
</head>
<body onload="welcome()" onunload="goodbye()">
</body>
</html>
```

13. Write a Javascript to design a form to accept values for user ID & password.

```
<html>
<body>
<form name="login">
```

```
Enter Username<input type="text" name="userid"><br>
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>
```

14. State any two properties and methods of location object.

Properties of location object:

1. hash
2. host
3. hostname
4. href
5. origin
6. pathname
7. port
8. protocol
9. search

Methods of location object:

1. assign ()

2. reload ()

3. replace ()

15. Explain i>Object name ii> Property

Object Name:

A JavaScript object is a collection of named values.

These named values are usually referred to as properties of the object.

Property:

A Property is a value or set of values that is the member of an object.

Example:

```
var person = new Object();
person.firstName = “Hhh";
person.age = 10;
```

In above example,

Person is an object and firstname and age are properites.

16. What is operator? Which type of operators are used in Java script.

JavaScript operators are symbols that are used to perform operations on operands.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators
7. Dot operator
8. Delete operator
9. typeof operator

10.new opearor
11.Conditional Operator(?:)
12.in operator

17. Write Java script to display current date and time.

```
<html>
        <body>
        <script>
         var today = new Date();
          var date = today.getFullYear()+'-'+(today.getMonth()+1)+'-'+today.getDate();
          var time = today.getHours() + ":" + today.getMinutes() + ":" +
        today.getSeconds();
          var dateTime = date+' '+time;
          document.write(dateTime);
        </script>
        </body>
        </html>
```

Output:

2023-1-12 14:54:26

18. Write a Java script to changing case of string

```
<script>
var text = "Hello World!";
var result = text.toLowerCase();
var res = text.toUpperCase();
document.write(result);
document.write("<br>"+res);
</script>
```

Output:

hello world!

HELLO WORLD

19.Explain Javascript method to find Unicode of a character

charCodeAt( )	• It provides the Unicode value of a character present at the specified index. • The index of the first character is 0, the second is 1, ... • Syntax: string.charCodeAt(*index*) • example:

	<script>
	var text = "HELLO WORLD";
	var c = text.charCodeAt(1);
	document.write(c);
	document.write("
	"+text.charCodeAt(7));
	document.write("
	"+text.charCodeAt(8));
	</script>

20. Explain following form control/elements with example. Button and checkbox

Button:

Defines a clickable button (mostly used with a JavaScript to activate a script) and is created by using following code:

```
<form method = “GET” action = “”>
<input type = “button” name = “MyButton” value = “Click” onclick = “msg()”>
<form>
```

A Button object also represents an HTML <button> element which is specified as follows:

```
<button name = “btn” value = “MyButton” onclick = “msg()”>
```

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

21. State and explain what a session cookie is.

The session cookie is a server-specific cookie that cannot be passed to any machine other than the one that generated the cookie. The session cookie allows the browser to re-identify itself to the single, unique server to which the client had previously authenticated.

22. State the features of JavaScript.

1) Lightweight

JavaScript does not have too many language constructs.
JavaScript uses dynamic typing, so everything that you declare or assign, the interpreter tries to figure out, what should be the type of a certain variable.

2) Object-oriented
3) Interpreted based
4) Handling date and Time
5) Validating user inputs
6) Event Handling

23. Differentiate between session cookies and persistent cookies.

Session Cookies	persistent cookies
It resides in memory for the length of the browser session.	A persistent cookie is a cookie that is assigned an expiration date.

Also known as an in-memory cookie	Also known as transient cookie.
Session cookie is automatically deleted when the user exits the browser application.	It is written to the computer's hard disk and remains there until the expiration date has been reached; then it's deleted.

Example of session cookie:

```
<script>
function createCookie()
{
var x = document.getElementById('myname').value
document.cookie = "name=" + x + ";"
alert("Cookie Written..")
}
</script>
```

Example of persistent cookie:

```
<script>
var date = new Date();
var days=2;
date.setTime(date.getTime()+(days*24*60*60*1000));
var expires = date.toGMTString();
document.cookie = "user=Rahul; expires="+ expires + ";"
alert("Cookie Created\n"+document.cookie)
</script>
```

24. Write a javascript program to check whether entered number is prime or not.

```
<html>
<body>
<script>
var i, chk=0;
```

```
var num = parseInt(prompt("Enter a positive number: "));
for(i=2; i<num; i++)
{
 if(num%2==0)
 {
  chk++;
  break;
 }
}
if(chk==0)
 document.write(num + " is a Prime Number");
else
 document.write(num + " is not a Prime Number");
</script>
</body>
</html>
```

25. Explain following form events : (i) onmouseup (ii) onblur

The onmouseup event occurs when **a mouse button is released over an element**.

Example:

```
<p id="myP" onmousedown="mouseDown()"
onmouseup="mouseUp()">
The mouseDown() function sets the color of this text to red.
The mouseUp() function sets the color of this text to blue.
</p>
```

The onblur event occurs when an **HTML element loses focus**.

Example:

```
<input type="text" id="fname" onblur="myFunction()">
<script>
function myFunction() {
  let x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
```

```
}
```

26. Write a javascript program to changing the contents of a window.

In following example, we are creating only one object of window and each time same window remain open and content of window changes.

```
<html>
<body>
<script>
function openWin1(ad)
{
  myWindow = window.open(ad, "myWindow",
"width=500,height=500");
}
</script>
<button value="Google"
onclick="openWin1('https://www.google.com')">Google</button>
<button value="Vidyalankar"
onclick="openWin1('http://vpt.edu.in')">Vidyalanakr</button>
</body>
<html>
```

27. Explain frame works of javascript and its application.

JavaScript frameworks are a type of tool that makes working with JavaScript easier and smoother.

1) Angular

2) React

3) Vue.js

4) Node.js

5) Backbone.js

28.Write a javascript syntax to accessing elements of another child window.

From the top level window that contains the iFrame you start by getting a reference to the iFrame by using plain old getElementById to select the iFrame

by Id. Alternately you can also access the frame via window.frames[0] (or appropriate numeric index). Once the iFrame is selected you can use the contentDocument property to access the child frame content. From there you can access document methods as you normally would – in this case by using getElementById() and then assigning some HTML to the display tag in the child frame.

Example: window.frames[0].showMessage("Hello from Main Page in iFrame");

4 Marks Questions

1. Explain getter and setter properties in Java script with 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.

```
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>
<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>
```

2.Explain prompt() and confirm() method of Java script with syntax and example.

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)

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>
```

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>
```

3.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

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("<br>");
 if (avg < 60){
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

4.Write the use of chatAt() and indexof() with syntax and example

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';
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

5.Differentiate between concat() and join() methods of array object.

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'

6. Write a JavaScript that will replace following specified value with another value in string. String = “I will fail” Replace “fail” by “pass”.

<html>

<head>

<body>

```
<script>
var myStr = 'I will fail';
var newStr = myStr.replace(fail, "pass");
document.write(newStr);
</script>
</body>
</head>
</html>
```

7.Write a Java Script code to display 5 elements of array in sorted order.

```
<html>
<head>
<title> Array</title>
</head>
<body>
<script>
var arr1 = [ "Red", "red", "Blue", "Green"]
document.write("Before sorting arra1=" + arr1);
document.write("<br>After sorting arra1=" + arr1.sort());
</script>
</body>
</html>
```

8. Explain open() method of window object with syntax and example.

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>
```

9. Describe regular expression. Explain search () method used in regular expression with suitable example.

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 + '<br>');

 // searching string without modifier i
 var n = str.search(/Morning/);

 document.write(n);
}
myFunction();
</script>
</body>
</html>
```

10. List ways of protecting your web page and describe any one of them.

Ways of protecting Web Page:

1)Hiding your source code

2)Disabling the right MouseButton

3) Hiding JavaScript

4) Concealing E-mail address.

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 file 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 file.

<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 file 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>
```

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

```
<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">
<br>
<input type="button" value="Next" onclick="slideshow(1)">
<input type="button" value="Back" onclick="slideshow(-1)">
</body>
</html>
```

12. Explain text rollover with suitable example

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>
</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">
</a>
</TD>
<TD>
<IMG height="1" src="" width="10">
</TD>
<TD>
<A onmouseover= "document.cover.src='sunflower.jpg'">
<B><U>Sunflower</U></B>
</A>
<BR>
<A onmouseover=
```

```
"document.cover.src='jasmine.jpg'">
<B><U>Jasmine</U></B>
</A>
<BR>
<A onmouseover=
"document.cover.src='rose.jpg'">
<B><U>Rose</U></B>
</A>
</TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>
```

13. 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.

```
<html>
<head>
<title>JavaScript Status Bar</title></head>
<body>
 <a href=" https://msbte.org.in/"
 onMouseOver="window.status='MSBTE';return true"
 onMouseOut="window.status='';return true">
```

```
MSBTE
</a>
</body>
</html>
```

14. Write a JavaScript program that will display current date in DD/MM/YYYY format.

```
<!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)
</script>
</body>
</html>
```

15. Write a JavaScript program that will remove the duplicate element from an array.

```
<!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>
```

16.Write a JavaScript program that will display list of students in ascending order according to the marks & calculate the average performance of the class.

Student Name	Marks
Amit	70
Sumit	78
Abhishek	71

```
<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>
```

17. Write and explain a string functions for converting string to number and number to string.

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.

```
Egvar a=prompt('Enter a number');

var b=parseInt(prompt('Enter a number'));

document.write(typeof a+"<br>");

document.write(typeof b);
```

To convert form number to string we can use toString()

```
<html>

<body>

<p>toString() returns a number as a string:</p>

<script>

let num = 12;

let text = num.toString();

document.write(num)

</script>

</body>

</html>
```

18. Differentiate between concat( ) & join( ) methods of array object.

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>

19. Write a JavaScript function to check the first character of a string is uppercase or not.

```
<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>
```

20.Write a JavaScript function to merge two array & removes all duplicate values.

```
<html>
<body>
<script>
function merge_array(array1, array2)
 {
 var result_array = [];
 var arr = array1.concat(array2);
 var len = arr.length;
 var assoc = {};
 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

21. Write a JavaScript function that will open new window when the user will click on the button.

```
<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("<p>Hello Everyone.Welcome to new window.</p>");
}
</script>
```

```
</body>

</html>
```

22. Describe text Rollover with the help of example.

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 fromthe text area

The HTML script is shown in the following example:

Example:

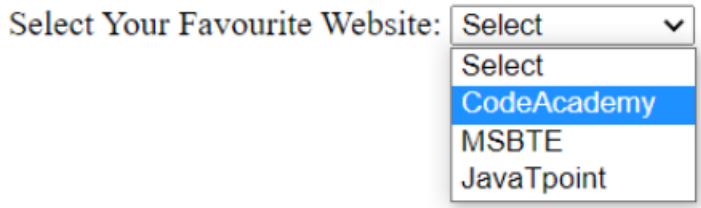
```
<html>

<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>
```

23. 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 re directed to that website.

```
<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>
```

Output:



24. Describe Quantifiers with the help of example.

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.

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>
```

25. Describe frameworks of JavaScript & its application.

Frameworks of JavaScript:

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:

1. JQuery can be used to develop Ajax based applications.

2. It can be used to make code simple, concise and reusable.

3. It simplifies the process of traversal of HTML DOM tree.

4. 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

26. Describe how to link banner advertisement to URL with example.

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" /></a>
</center>
</body> </html>
```

27. Explain alert method in Java script with suitable example.

The alert() method displays an alert box with a message and an OK button.

The alert() method is used when you want information to come through to the user.

Syntax:
alert(*message*)
example:

```
<html>
<body>
<h1>The Window Object</h1>
<h2>The alert() Method</h2>
<p>Click the button to see line-breaks in an alert box.</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction()
 {
  alert("Hello\nHow are you?");
}
</script>
</body>
</html>
```

28. Explain getter and setter properties in Java script with suitable example.

Also known as Javascript assessors.

Getters and setters allow you to control how important variables are accessed and updated in your code.

JavaScript can secure better data quality when using getters and setters.
example: Getters and setters allow you to get and set properties via methods.

```
<script>
var person = {
firstName: 'Chirag',
lastName: 'Shetty',
get fullName()
{
return this.firstName + ' ' + this.lastName;
},
set fullName (name)
{
var words = name.split(' ');
this.firstName = words[0];
this.firstName = words[0].toUpperCase();
this.lastName = words[1];
}
}
document.write(person.fullName); //Getters and setters allow you to get and set properties via methods.
document.write("<br>"+"before using set fullname()"+"<br>");
person.fullName = 'Yash Desai'; //Set a property using set
document.writeln(person.firstName); // Yash
document.write(person.lastName); // Desai
</script>
```

Output:

Chirag Shetty before using set fullname() YASH Desai

29. Write a Javascript that displays the following properties of Math object.

```
<script type="text/javascript">
document.write("square root of 16 is =" + Math.sqrt(16) + "<br>");
document.write("floor of 16.877 is=" + Math.floor(16.877) + "<br>");
document.write("ceil of 16.877 is=" + Math.ceil(16.877) + "<br>");
```

```
document.write("ceil of 16.241 is=" + Math.ceil(16.241) + "<br>");
document.write("sin of 90 is=" + Math.sin(90));
</script>
```

Output:

square root of 16 is =4
floor of 16.877 is=16
ceil of 16.877 is=17
ceil of 16.241 is=17
sin of 90 is=0.8939966636005579

30. Write a javascript function that concatenate the two strings.

```
<script>
function strcon(p1, p2)
{
  return p1.concat(p2);
}
document.write(strcon("information","technology"));
</script>
```

Output:

Informationtechnology

31. Differentiate between concat() and join() methods of array object.

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>

32. Write a program to changing case of string.

```
<script>
var text = "Hello World!";
var result = text.toLowerCase();
var res = text.toUpperCase();
document.write(result);
document.write("<br>"+res);
</script>
```

Output:

hello world!

HELLO WORLD

33. How to add and sort elements in array? Explain with example.

The **array.sort()** is an inbuilt method in JavaScript which is used to sort the array.

Syntax:

array.sort();

Here array is the set of values which is going to be sorted.

Example:

```
<script>
var arr1 =["Red", "red", 200,"Blue",100];
document.write("Before sorting arra1=" + arr1);
document.write("<br>After sorting arra1="+ arr1.sort());
</script>
```

Output:

Before sorting arra1=Red,red,200,Blue,100
After sorting arra1=100,200,Blue,Red,red

Adding elements into an array:

Method1:

The easiest way to add a new element to an array is using the push() method. The push() method adds new items to the end of an array, and returns the new length.

Syntax:

array.push(*item1, item2, ..., itemX*);

Example:

```
var fruits = [ "Banana", "Orange", "Apple", "Mango" ];
fruits.push( "Lemon" );    // adds a new element (Lemon) to fruits
```

Method 2:

The unshift() method adds one or more elements to the beginning of an array and returns the new length of the array.

Syntax:

array.unshift(*item1, item2, ..., itemX*);

Example:

```
var fruits = [ "Banana", "Orange", "Apple", "Mango" ];
fruits.unshift( "Lemon","Pineapple" );
```

Method 3:

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+"<br>");
fruits[fruits.length] = "Kiwi";
document.write(fruits+"<br>");
fruits[fruits.length] = "Chikoo";
document.write(fruits);
</script>
```

Output:

```
Banana,Orange,Apple,Mango
Banana,Orange,Apple,Mango,Kiwi
Banana,Orange,Apple,Mango,Kiwi,Chikoo
```

Method 4:

The **splice()** 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);
```

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).

The **splice()** 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);

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).

Example:

```
<script>
var fruits = ["Banana", "Watermelon", "Chikoo", "Mango", "Orange", "Apple"];
document.write(fruits+"<br>");
fruits.splice(2,2, "Lemon", "Kiwi");
document.write(fruits+"<br>");
fruits.splice(0,2); //removes first 2 elements from array
document.write(fruits+"<br>");
</script>
```

Output:

```
Banana,Watermelon,Chikoo,Mango,Orange,Apple
Banana,Watermelon,Lemon,Kiwi,Orange,Apple
Lemon,Kiwi,Orange,Apple
```

34. How to read and write cookie in javascript? Explain with example.

Read a Cookie with JavaScript

With JavaScript, cookies can be read like this:

```
var x = document.cookie;
```

Write a Cookie with JavaScript

You can access the cookie like this which will return all the cookies saved for the current domain.

document.cookie=x;

Example:

```
<html>
<head>
<script>
function writeCookie()
{
with(document.myform)
{
document.cookie="Name=" + person.value + ";"
alert("Cookie written");
}
}

function readCookie()
{
var x;
if(document.cookie=="")
x="";
else
x=document.cookie;
document.write(x);
}
</script>
</head>
<body>
<form name="myform" action="">
```

```
Enter your name:
<input type="text" name="person"><br>
<input type="Reset" value="Set Cookie" type="button" onclick="writeCookie()">
<input type="Reset" value="Get Cookie" type="button" onclick="readCookie()">
</form>
</body>
</html>
```

35. How to finding non matching characters in regular expression? Explain with suitable example.

The [^abc] expression is used to find any character NOT between the brackets.

The characters inside the brackets can be any characters or span of characters:

- [abcde..] - Any character between the brackets
- [A-Z] - Any character from uppercase A to uppercase Z
- [a-z] - Any character from lowercase a to lowercase z
- [A-z]- Any character from uppercase A to lowercase z

- Sometimes a JavaScript application prohibits certain characters from appearing within text entered into a form, such as a hyphen (-); otherwise, the character might inhibit processing of the form by the CGI program running on the web server.
- You can direct the browser to search for illegal character(s) by specifying the illegal character(s) within brackets and by placing the caret (^) as the first character in the bracket.
- Let's see how this works in the following example:
  /[^\-]/
- In this case, the browser is asked to determine whether the text does not contain the hyphen.

- The caret asks the browser to determine whether the following character(s) do not appear in the text.

Syntax

```
new RegExp("[^xyz]")
or simply:
/[^xyz]/
```

Syntax with modifiers

```
new RegExp("[^xyz]", "g")
or simply:
/\[^xyz]/g
```

Example:

```
<html> <script>
function check()
{
var exp=/[^\*]/;
var res=exp.test(document.getElementById("txt1").value);
document.getElementById("demo1").innerHTML=res;
}
</script>
<body>
Enter text:<textarea id="txt1"></textarea>
<input type="button" onclick="check()" value="Check">
<p id="demo1"></p>
</body>
</html>
```

36. What is status bar? How to create status bar in Javascript.

The **status** property of the Window interface was originally intended to set the text in the status bar at the bottom of the browser window. However, the HTML

standard now requires setting window.status to have no effect on the text displayed in the status bar.

Syntax:

```
window.status = string;
var value = window.status;
```

```
<html>
<head>
<title>JavaScript Status Bar</title></head>
<body>
 <a href="http://www.v2v.edu.in"
  onMouseOver="window.status='Vision to Victroy';return true"
  onMouseOut="window.status='';return true">
Vision 2 Victory
 </a>
</body>
</html>
```

37. What is slideshow? Explain with example.

A slideshow is similar in concept to a banner advertisement in that a slideshow rotates multiple images on the web page. However, unlike a banner advertisement, a slideshow gives the visitor the ability to change the image that's displayed: the visitor can click the Forward button to see the next image and the Back button.
Example:

```
<html>
<title>slideshow</title>
<body>
<h2 class="w3-center">Manual Slideshow</h2>
```

```
<div class="w3">
  <img class="mySlides" src="1.jpg" style="width:50%">
  <img class="mySlides" src="2.jpg" style="width:50%">
  <img class="mySlides" src="3.jpg" style="width:50%">
  <img class="mySlides" src="4.jpg" style="width:50%">

  <button class="aa" onclick="plusDivs(-1)">&#10094;Back</button>
  <button class="bb" onclick="plusDivs(1)">&#10095;Forward</button>
</div>

<script>
var slideIndex = 1;
showDivs(slideIndex);

function plusDivs(n)
{
  showDivs(slideIndex += n);
}

function showDivs(n)
 {
  var i;
  var x = document.getElementsByClassName("mySlides");
  if (n > x.length)
            {
            slideIndex = 1
            }

  if (n < 1)
            {
            slideIndex = x.length
            }
  for (i = 0; i < x.length; i++)
            {
            x[i].style.display = "none";
            }
```

```
  x[slideIndex-1].style.display = "block";
}
</script>
</body>
</html>
```

38. Describe how to replace text using Regular Expression with example.

- you can also use a regular expression to replace portions of the text by using the replace() method.
- The replace() method requires two parameters: a regular expression and the
- replacement text.

The replace() method searches a string for a specified value, or a *regular exp*

Syntax

```
string.replace(searchvalue, newvalue)
```

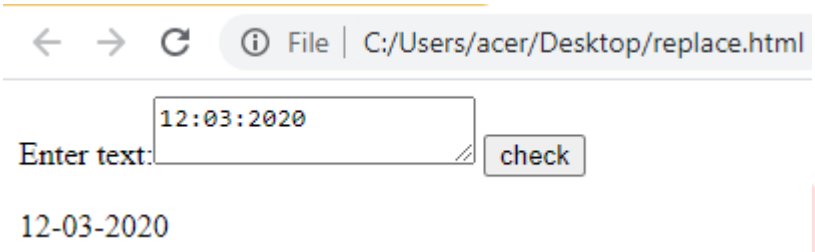
Example:

```
<html>
<script>
function check()
{
var exp=/:/g;
var str=document.getElementById("txt").value;
var res=str.replace(exp,"-");
document.getElementById("demo1").innerHTML=res;
}
</script>
<body>
Enter text:<textarea id="txt"></textarea>
<input type="button" onclick="check()" value="check">
<p id="demo1"></p>
</body>
```


80

```
</html>
```

Output:



In the above example ":" symbol replace by "-"

39. What is context menu? How to create it? Explain with example.

When we click the right mouse button on our desktop, a menu-like box appears and this box is called the context menu. In JavaScript, a context menu event runs when a user tries to open a context menu. This can be done by clicking the right mouse button.

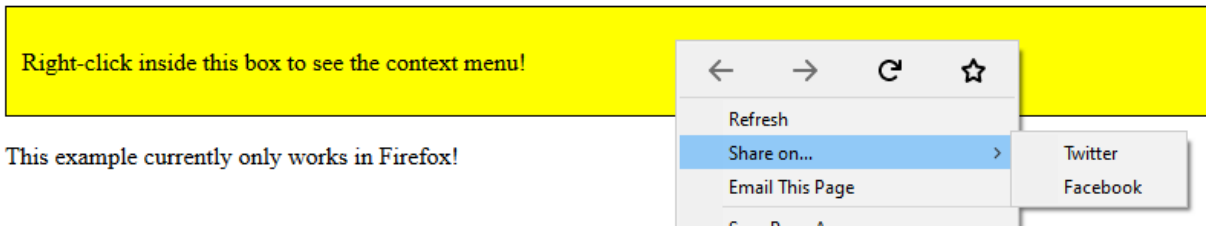
Example:

```
<html>
<head>
<style>
div {
  background: yellow;
  border: 1px solid black;
  padding: 10px;
}
</style>
</head>
<body>

<div contextmenu="mymenu">
<p>Right-click inside this box to see the context menu!
```


81

```
<menu type="context" id="mymenu">
<menuitem label="Refresh" onclick="window.location.reload();"
icon="ico_reload.png"></menuitem>
  <menu label="Share on...">
    <menuitem label="Twitter" icon="ico_twitter.png"
onclick="window.open('//twitter.com/intent/tweet?text=' +
window.location.href);"></menuitem>
    <menuitem label="Facebook" icon="ico_facebook.png"
onclick="window.open('//facebook.com/sharer/sharer.php?u=' +
window.location.href);"></menuitem>
  </menu>
  <menuitem label="Email This Page"
onclick="window.location='mailto:?body='+window.location.href;"></men
uitem>
</menu>

</div>

<p>This example currently only works in Firefox!</p>

</body>
</html>
```

Output:



82

40. Write a javascript program to validate user accounts for multiple set of user ID and password (using swith case statement).

```
<html>
<body>
Enter your User ID
<input type="text" id="id">
<br><br>
Enter your Password
<input type="password" id="pass">
<br><br>
<input type="submit" onclick="check()">
<br><br>
<p id="display"></p>
<script>
function check() {
var uid = document.getElementById('id').value;
var pass = document.getElementById('pass').value;
switch(uid){
case "darshan.khapekar@vpt.edu.in":
if(pass == "darshan@123"){
document.getElementById('display').innerHTML = "Valid User";
}
break;
case "prashant.yelurkar@vpt.edu.in":
if(pass == "prashant@123"){
document.getElementById('display').innerHTML = "Valid User";
}
break;
case "konisha.thakare@vpt.edu.in":
```

```
if(pass == "konisha@123"){
document.getElementById('display').innerHTML = "Valid User";
}
break;
default:
document.getElementById('display').innerHTML = "Invalid User";
}
}
</script>
</body>
</html>
```

41. Differentiate between concat() and join() methods of array object.

concat() of array object	join() of array object
The concat() method concatenates (joins) two or more arrays.	The join() method returns an array as a string.
The concat() method does not change the existing arrays.	The join() method does not change the original array.
Syntax: array1.concat(*array2*, *array3*, ..., *arrayX*)	Syntax: array.join(*separator*) Any separator can be specified. The default is comma (,).

Example:
<script>
var arr1 = ["Java", "Python"];
var arr2 = [1, 2, 3];
var arr3 = arr1.concat(arr2);
document.write(arr3);
</script>
	Example:
	<script>
	var fruits = ["Banana", "Orange", "Apple", "Mango"];
	var text = fruits.join("#");
	document.write(text);
	</script>

42.Write a javascript program to demonstrate java intrinsic function.

An intrinsic function is often used to replace the Submit button and the Reset button with your own graphical images, which are displayed on a form in place of these buttons.

Example:

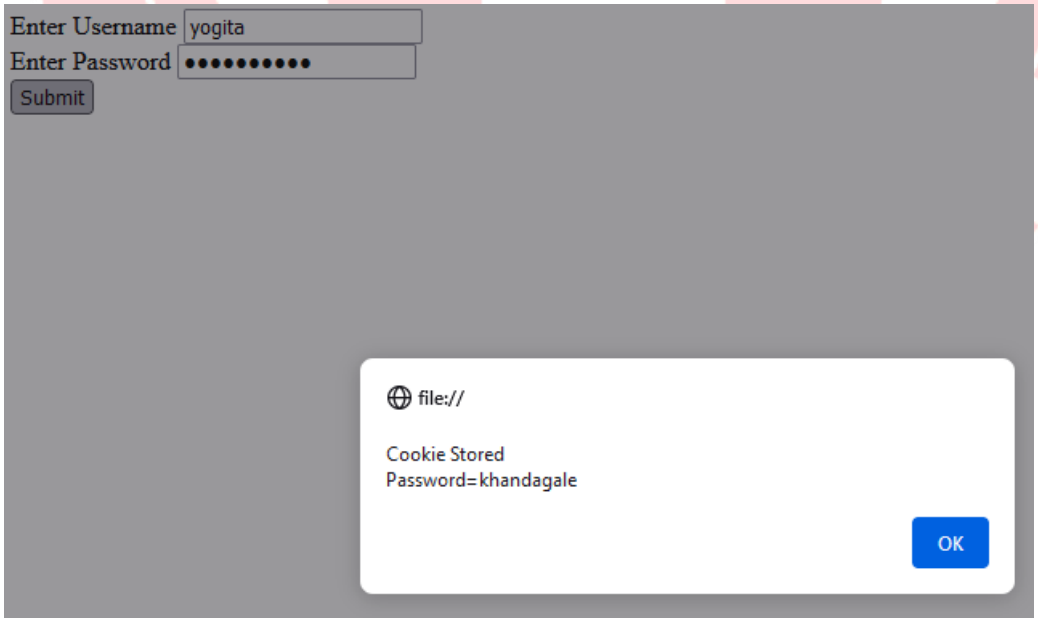
```
<html>
<head>
<title>Using Intrinsic JavaScript Functions</title>
</head>
<body>
 <FORM name="contact" action="#" method="post">
  <P>
  First Name: <INPUT type="text" name="Fname"/> <BR>
  Last Name: <INPUT type="text" name="Lname"/><BR>
  Email: <INPUT type="text" name="Email"/><BR>
  <img src="submit.jpg"
          onclick="javascript:document.forms.contact.submit()"/>
  <img src="reset.jpg"
          onclick="javascript:document.forms.contact.reset()"/>
  </P>
 </FORM>
</body>
</html>
```

43. Design 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 become value of the cookies. Write the javascript function for storing the cookies.

Code:

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

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



44. Write a javascript program to create read, update and delete cookies.

Code:

```
<html>
<head>
<script>
function writeCookie()
{
var d=new Date();
d.setTime(d.getTime()+(1000*60*60*24));
with(document.myform)
{
document.cookie="Name=" + person.value + ";expires="
+d.toGMTString();
}
}
function readCookie()
{
if(document.cookie=="")
document.write("cookies not found");
else
document.write(document.cookie);
}
</script>
</head>
<body>
<form name="myform" action="">
Enter your name:
<input type="text" name="person"><br>
```

```
        <input type="Reset" value="Set" type="button"
onclick="writeCookie()">
        <input type="Reset" value="Get" type="button"
onclick="readCookie()">
        </form>
        </body>
        </html>
```

45. Write a javascript program to link banner advertisements to different URLs.

Code:

```
<html>
<head>
<title>Link Banner Ads</title>
<script language="Javascript" type="text/javascript">

Banners = new Array('y.jpg','yy.jpg','yyy.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="200" name="RotateBanner" /></a>
</center>
</body>
</html>
```

46. Write a javascript program to calculate add, sub, multiplication and division of two number (input from user). Form should contain two text boxes to input numbers of four buttons for addition, subtraction, multiplication and division.

Code:

```
<html>
<head>
<script>
function multiplyBy()
{
        num1 = document.getElementById("firstNumber").value;
        num2 = document.getElementById("secondNumber").value;
        document.getElementById("result").innerHTML = num1 * num2;
}

function divideBy()
{
```

```
        num1 = document.getElementById("firstNumber").value;
        num2 = document.getElementById("secondNumber").value;
document.getElementById("result").innerHTML = num1 / num2;
}
function add()
{
        num1 = parseInt(document.getElementById("firstNumber").value);
        num2 =
parseInt(document.getElementById("secondNumber").value);
        document.getElementById("result").innerHTML = num1 + num2;
}
function subtract()
{
        num1 = document.getElementById("firstNumber").value;
        num2 = document.getElementById("secondNumber").value;
        document.getElementById("result").innerHTML = num1 - num2;
}
</script>
</head>
<body>
<form>
1st Number : <input type="text" id="firstNumber"><br>
2nd Number: <input type="text" id="secondNumber"><br><br>
<input type="button" onClick="multiplyBy()" Value="Multiply">
<input type="button" onClick="divideBy()" Value="Divide">
<input type="button" onClick="add()" Value="Addition">
<input type="button" onClick="subtract()" Value="Subtraction">
</form>
<p>The Result is : <br>
<p id = "result"></p>
</p>
```

</body>
</html>

47. State what is regular expression. Explain its meaning with the help of a suitable example.

A regular expression is an object that describes a pattern of characters.

The JavaScript **RegExp** class represents regular expressions, and both String and **RegExp** define methods that use regular expressions to perform powerful pattern-matching and search-and-replace functions on text.

A **Regular Expression** is a sequence of characters that constructs a search pattern. When you search for data in a text, you can use this search pattern to describe what you are looking for.

Syntax:
A regular expression is defined with the **RegExp ()** constructor as:

```
var pattern = new RegExp(pattern, attributes);
```

or simply

```
var pattern = /[pattern]/attributes;
```

Here,

- **Pattern –** A string that specifies the pattern of the regular expression or another regular expression.
- **Attributes –** An optional string containing attributes that specify global, case-insensitive, and multi-line matches.

Example:

```
<html>
```

```
<body>
<script>
//validating mobile number
function validatePhone(num) {

    // regex pattern for phone number
    var re =/^\d{10}$/;

    // check if the phone number is valid
    var result = num.match(re);
    if (result) {
        document.write('The number is valid.');
    }
    }
// take input
var number = prompt('Enter a 10 digit mob number ');
validatePhone(number);
</script>
</body>
</html>
```

48. Differentiate between For-loop and For-in loop.

For loop	For-in loop
for loop provides a concise way of writing the loop structure.	For-in loop in JavaScript is used to iterate over the properties of an object.
For instance, if you want to iterate over *even* numbers, you'd need to use the normal for loop	Certainly, for objects, the for-in loop allows you to get the property name in the iteration variable.

for (*expr 1*; *expr 2*; *expr3*)
{
  // code block to be executed
}
	for (x in object) {
	  code block to be executed
	}

// program to display text 5 times
const n = 5;
 	
// looping from i = 1 to 5
for (let i = 1; i <= n; i++) {
  console.log(`I love JavaScript.`);
}
	<script>
	var code= {
	  CO : "Comp Engg.",
	  IF : "Info Tech",
	  EJ : "Electronics"
	}
	// using for...in
	for (var i in code) {
	  var c= code[i];
	  // display the values
	
	document.write(i+"="+c+"
	");
	}
	</script>

49. Write a javascript function that accepts a string as a parameter and find the length of the string.

Code:

```
<html>
<body>
<p id="demo"></p>
<script>
function len(text)
{
  return (text.length);
}
document.getElementById("demo").innerHTML = "length of
string="+len("Information");
```

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

Output:

```
length of string=11
```

50. Write a javascript program to validate email ID of the user using regular expression.

Code:

```
<html>
<head>
<title>JavaScript Regular expression to valid an email address</title>
</head>
<body>
<script>
function valid_email(str)
{
var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(mailformat.test(str))
{
alert("Valid email address!");
}
else
{
alert("You have entered an invalid email address!");
}
}
valid_email('yogita.khandagale@gmail.com');
</script>
</body>
```

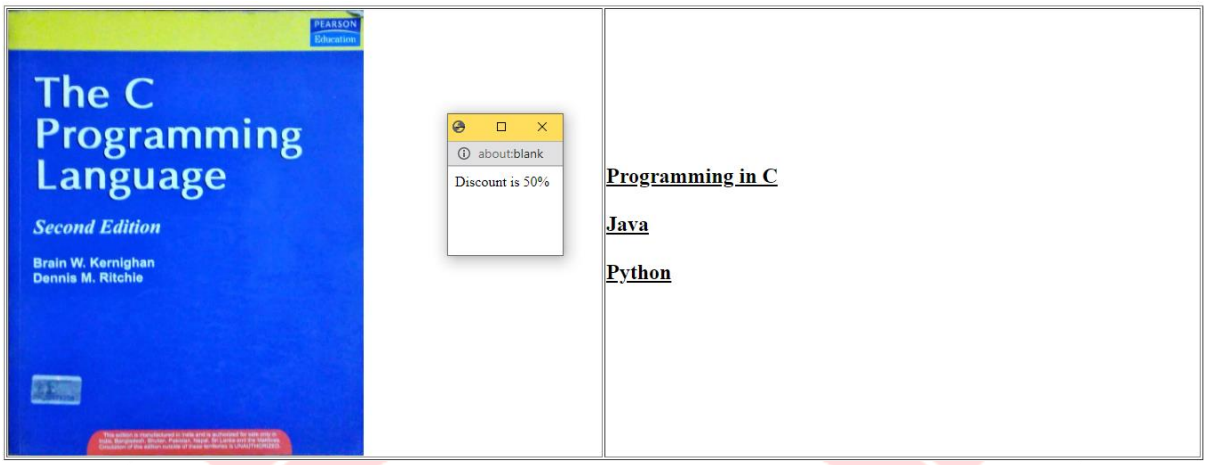
```
</html>
```

51. Write a javascript program to design HTML page with books information in tabular format, use rollovers to display the discount information.

```
<html>
<head>
<title>
table rollovers</title>
<script>
function open_new_window(clrname)
{
if(clrname==1)
{
document.clr.src="c.jpg";
mwin=window.open('','myadwin','height=50,width=50,left=500,top=200');
mwin.document.write("Discount is 50%");
}
if(clrname==2)
{
document.clr.src="java.jpg";
mwin=window.open('','myadwin','height=50,width=50,left=500,top=200');
mwin.document.write("Discount is 40%");
}

if(clrname==3)
{
document.clr.src="p.jpg";
mwin=window.open('','myadwin','height=100,width=150,left=500,top=200');
mwin.document.write("Discount is 20%");
}
}
</script>
</head>
<body>
```

```
<table border="1" width="100%">
<tbody>
<tr>
<td width="50%">
<a><img height="500" src="c.jpg" width="400" name="clr"></td>
<td><H2>
<a onmouseover="open_new_window(1)" onmouseout="mwin.close()">
<b><u>Programming in C</u></b></a>
<br><br>
<a onmouseover="open_new_window(2)" onmouseout="mwin.close()">
<b><u>Java</u></b></a>
<br><br>
<a onmouseover="open_new_window(3)" onmouseout="mwin.close()">
<b><u>Python</u></b></a>
</H2>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Output:



52. List ways of protecting your webpage and describe any one of them.

There is nothing secret about your web page. Anyone with a little computer knowledge can use a few mouse clicks to display your HTML code, including your JavaScript, on the screen.

Following are the ways to protect web pages:

1) Hiding Your Code by disabling Right Mouse Click:

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.

```
<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.

For example, this can be useful when:

- Clicking on a "Submit" button, prevent it from submitting a form
- Clicking on a link, prevent the link from following the URL

2) 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.

webpage.html

```
<html>
<head>
<script src="mycode.js" languages="javascript" type="text/javascript">
</script>
<body>
<h3> Right Click on screen, Context Menu is disabled</h3>
</body>
</html>
```

mycode.js

```
window.onload=function()
{
document.addEventListener("contextmenu", function(e)
{
```

```
e.preventDefault();
}, false);
}
```

3) Concealing Your E-mail Address

- 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.

```
<html>
<head>
<title>Conceal Email Address</title>
<script>

function CreateEmailAddress()
```

```
{
var x = 'abcxyz*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="send" onclick="CreateEmailAddress()">
</body>
</html>
```

6 MARKS QUESTIONS

1. 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.

```
<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>
 <select name="optionList" size="2">
<option value=1>Mango
<option value=2>Banana
<option value=3>Apple
 </select>
 <br>
 <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
 <br>
 <input name="Reset" value="Reset" type="reset">
 </p>
</form>
</body>
</html>
```

2. Describe, how to read cookie value and write a cookie value. Explain with example.

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.

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>
```

3. 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.

```
<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 == "" ) {
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>
```

4. Describe how to evaluate checkbox selection. Explain with suitable example.

Evaluating Checkbox Selection:

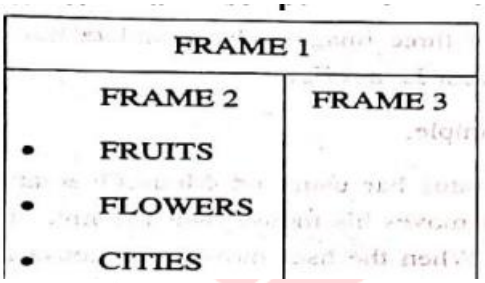
- A checkbox is created by using the input element with the type=”checkbox” attribute-value pair.
- A checkbox in a form has only two states(checked or un-checked) and is independent of the state of other checkboxes in the form. Check boxes can be grouped together under a common name.
- You can write 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>
```

5.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.




```
<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
<ul>
<li>
<a href="fruits.html" target="mainframe">FRUITS</a>
```

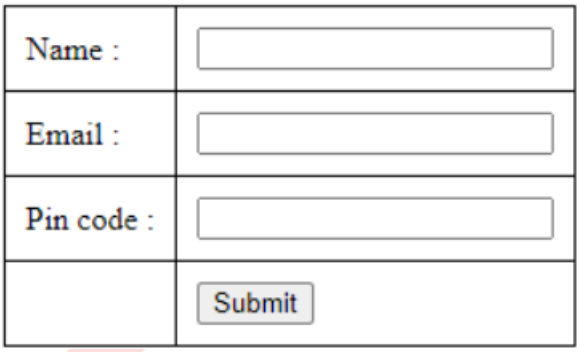
```
</li>
<li>
<a href="flowers.html" target="mainframe">FLOWERS</a>
</li>
<li>
<a href="cities.html" target="mainframe">CITIES</a>
</li>
</ul>
</td>
<td>
FRAME 3<BR>
<iframe name="mainframe"></iframe>
</td>
</tr>
</table>
</body>
</html>
```

6. 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.

```
<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>
```

7. 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.




```
<html>
<head>
<style>
table,tr,td
{
border: solid black 1px;
border-collapse: collapse;
}

td
{
padding: 10px;
}
</style>
```

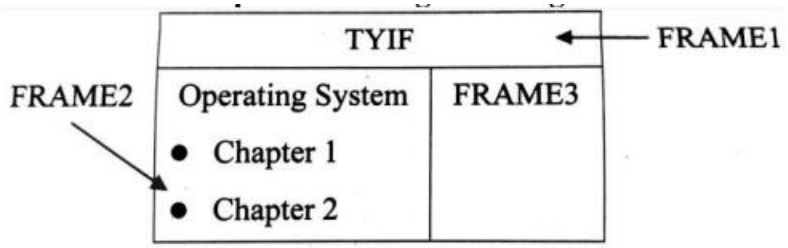
```
</head>
<body>
<table>
<tbody>
<tr>
<td>Name : </td>
<td> <input type="text" id="name" required></td>
</tr>
<tr>
<td>Email : </td>
<td> <input type="email" id="email" required></td>
</tr>
<tr>
<td>Pin code : </td>
<td> <input type="number" id="pin" required></td>
</tr>
<tr>
<td></td>
<td><button 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>
```

8.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.

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

9. Write a JavaScript for creating following frame structure: 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.



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>
```

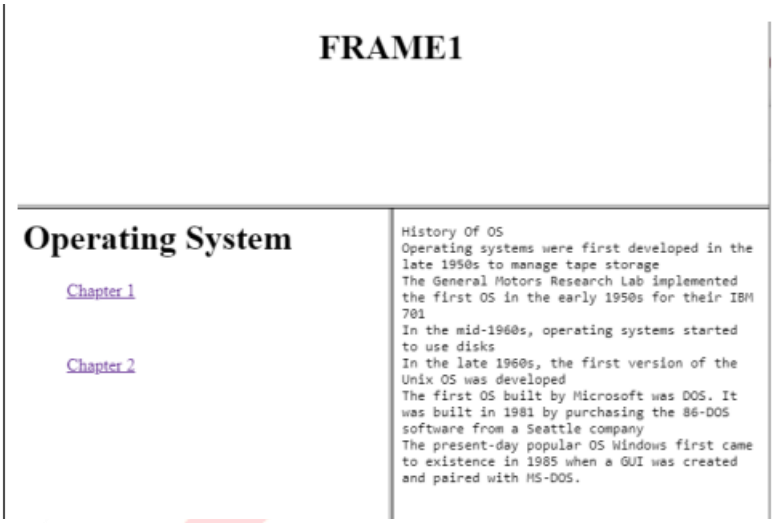
Step 3) create frame3.html

```
<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:



10. 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.

```
<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>
```

11. Develop a JavaScript program to create Rotating Banner Ads.

```
<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;
}
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>
```

12. Write a JavaScript for the folding tree menu.

```
<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>
<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>
```

13. Write a HTML script that displays textboxes for accepting emp id, name and designation. Write proper JavaScript such that when the user clocks the submit button

i) all textboxes change the color to Blue

ii) constructs mail id as <id><name>@yahoo.com and display mail id.

```
<html>
```

```
<body>
<form action=" ">
EmpId:
<input type="text" id="eid" onchange="highlightid(this)">
<br><br>
Name:
<input type="text" id="name" onchange="highlightname(this)">
<br><br>
Designation: <input type="text" id="des"
onchange="highlightdes(this)">        <br><br>
</form>
<p>Click the button to get the details:</p>
<button onclick="myFunction()">Details</button>
<BR>
<p id="demo"></p>
<p id="demo1"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<script>
function myFunction()
{
 var y = document.getElementById("eid").value;
 document.getElementById("demo").innerHTML = y;
 var x = document.getElementById("name").value;
 document.getElementById("demo1").innerHTML = x;
 var z = document.getElementById("des").value;
 document.getElementById("demo2").innerHTML = z;
var g=y.concat(x);
var k=g.concat("@yahoo.com");
document.getElementById("demo3").innerHTML = k;
 }
function highlightid(y)
 {
y.style.color="blue";
 y.style.backgroundColor="pink";
 }
```

```
function highlightname(x)
    {
x.style.color="blue";
    x.style.backgroundColor="pink";
    }
function highlightdes(z)
    {
z.style.color="blue";
    z.style.backgroundColor="pink";
    }

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

Output:

EmpId: vp0179

Name: yogita

Designation: HOD

Click the button to get the details:

Details

vp0179

yogita

HOD

vp0179yogita@yahoo.com

14.Write a JavaScript for moving car using setTimeOut() and clearTimeOut() method.

```
<html>
<head>
<title>Animation </title>
<script type="text/javascript">
var obj=null;
var animate;
function init()
{
obj=document.getElementById('car');
obj.style.position='relative';
obj.style.left='0px';
}

function start()
{
obj.style.left=parseInt(obj.style.left)+ 10 + 'px';
animate=setTimeout(start,10);
}
```

```
function stop()
{
clearTimeout(animate);
obj.style.left='0 px';
}
window.onload=init;
</script>
</head>
<body>
<img id="car" src="car.jpg">

<br><br>
<input value="Start" type="button" onclick="start()"/>

<input value="Stop" type="button" onclick="stop()"/>
</body>
</html>
```

15. What is frame? How to create? Explain with example.

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

How to create frame?

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Following is the example to create three horizontal frames −

```
<html>
<head>
<title>Create a Frame</title>
```

```
</head>
<frameset rows="50%,30%,*">
<frame src="webpage1.html" name="topPage" />
<frame src="webpage2.html" name="bottomPage" />
<frame src="webpage3.html" name="bottomPage" />
</frameset>
</html>
```

Output:

16.Describe how to change option list dynamically with the help of example.

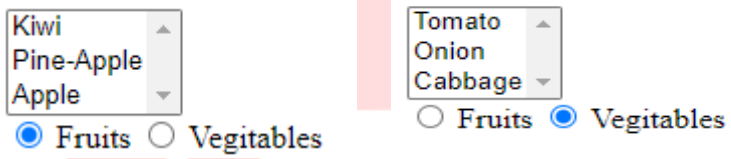
When user will select the fruits radio button, the option list should present only the fruits names to user and when user will select the vegetable radio button, the option list should present only the vegetable names to user so that user can select an appropriate element of interest.

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

```
<html>
<script type="text/javascript">
function modifyList(x)
{
with(document.forms.myform)
{
if(x ==1)
{
optionList[0].text="Kiwi";
optionList[0].value=1;
optionList[1].text="Pine-Apple ";
optionList[1].value=2;
optionList[2].text="Apple";
optionList[2].value=3;
}

if(x ==2)
{
optionList[0].text="Tomato";
optionList[0].value=1;
optionList[1].text="Onion ";
optionList[1].value=2;
optionList[2].text="Cabbage ";
optionList[2].value=3;
}
}
}

</script>
</head>
```

```
<body>
<form name="myform" action=" " method="post">
<select name="optionList" size="3">
<option value=1>Kiwi
<option value=1>Pine-Apple
<option value=1>Apple
</select>
<br>
<input type="radio" name="grp1" value=1 checked="true"
onclick="modifyList(this.value)"> Fruits

<input type="radio" name="grp1" value=2
onclick="modifyList(this.value)"> Vegitables
</form>
</body>
</html>
```

Output:



17.Write a javascript to create rotating banner ads with URL links..

Code:

```
<html>
<head>
<title>Link Banner Ads</title>
<script language="Javascript" type="text/javascript">

Banners = new Array('y.jpg','yy.jpg','yyy.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="200" name="RotateBanner" /></a>
</center>
</body>
</html>
```

18. Write a Javascript to create menu using “folding tree menu”.

Code:

```
<html>
<head>
<style>
ul, #myUL {
  list-style-type: none;
```

```
}

#myUL {
  margin: 0;
  padding: 0;
}

.caret {
  cursor: pointer;
  -webkit-user-select: none; /* Safari 3.1+ */
  -moz-user-select: none; /* Firefox 2+ */
  -ms-user-select: none; /* IE 10+ */
  user-select: 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>
<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>
```

Output:

Tree Menu

A tree menu represents a hierarchical view of information, where each item can have a number of subitems.

Click on the arrow(s) to open or close the tree branches.

▼ India
- Karnataka
- Tamilnaadu
- ▼ Maharashtra
  - Mumbai
  - Pune
  - ▼ Navi Mumbai
    - Nerul
    - Vashi
    - Panvel

19. Write a javascript to checks whether a passed string is palindrome or not.

```
<script>
// program to check if the string is palindrome or not
function checkPalindrome(string)
{
    // convert string to an array
    const arrayValues = string.split('');
    // reverse the array values
    const reverseArrayValues = arrayValues.reverse();
    // convert array to string
    const reverseString = reverseArrayValues.join('');

    if(string == reverseString) {
        document.write('It is a palindrome');
    }
    else {
        document.write('It is not a palindrome');
    }
```

```
}
//take input
const string = prompt('Enter a string: ');
checkPalindrome(string);
</script>
```

20. Develop javascript to convert the given character to unicode and vice-versa.

```
<script>
var x="HeLLO";
document.writeln(x.charCodeAt(3));
document.writeln("<br>"+x.charCodeAt(1));
document.writeln("<br>"+x.charCodeAt(4));
document.writeln("<br>");
var res = String.fromCharCode(72, 69, 76, 76, 79);
document.write(res);
</script>
```

Output:

```
76
101
79
HELLO
```

21. Write a javascript program to create a silde show with the group of six images, also simulate the next and previous transition between slides in your javascript.

```
<html>
<title>slideshow</title>
<body>
<h2 class="w3-center">Manual Slideshow</h2>
<div class="w3">
  <img class="mySlides" src="1.jpg" style="width:50%">
  <img class="mySlides" src="2.jpg" style="width:50%">
```

```
  <img class="mySlides" src="3.jpg" style="width:50%">
  <img class="mySlides" src="4.jpg" style="width:50%">
<img class="mySlides" src="5jpg" style="width:50%">
<img class="mySlides" src="6.jpg" style="width:50%">

  <button class="aa" onclick="plusDivs(-1)">&#10094;Back</button>
  <button class="bb" onclick="plusDivs(1)">&#10095;Forward</button>
</div>

<script>
var slideIndex = 1;
showDivs(slideIndex);

function plusDivs(n)
{
  showDivs(slideIndex += n);
}

function showDivs(n)
 {
  var i;
  var x = document.getElementsByClassName("mySlides");
  if (n > x.length)
            {
            slideIndex = 1
            }

  if (n < 1)
            {
            slideIndex = x.length
            }
```

```
  for (i = 0; i < x.length; i++)
          {
          x[i].style.display = "none";
          }
  x[slideIndex-1].style.display = "block";
}
</script>
</body>
</html>
```

22. Write a javascript to open a new window and the new window is having two frames. One frame containing buthon as “click here !”, and after clicking this button an image should open in the second frame of that child window.

Code:

f.html

```
<html>
<head>
<title>Create a Frame</title>
</head>
<frameset rows="60%,40%">
<frame src="b.html" name="a" />
<frame src="im.html" name="b" />
</frameset>
</frameset>
</html>
```

b.html

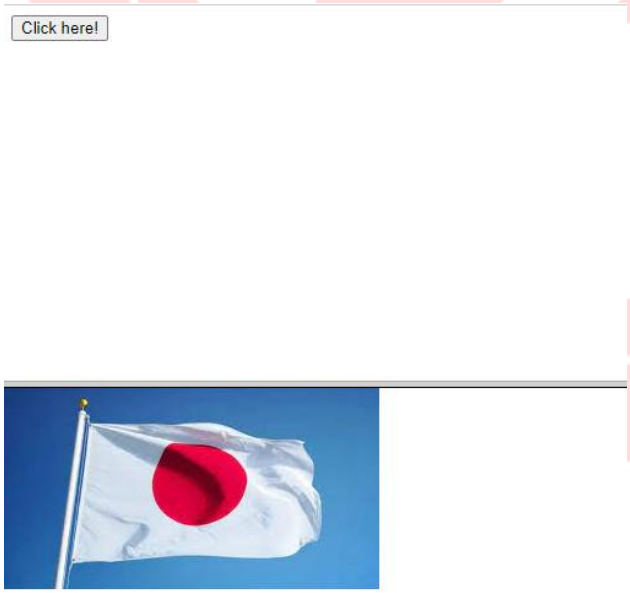
```
<html>
<head>
<title>Web Page 1</title>
</head>
<body>
<form action="" method="post">
```

```
<p><a href="j.png" target="b">
<input name="WebPage1" value="Click here!" type="button">
</a></p>
</body>
</html>
```

im.html

```
<html>
<head>
<title>Web Page 1</title>
</head>
<body>
<img src="i.png">
</body>
</html>
```

Output:



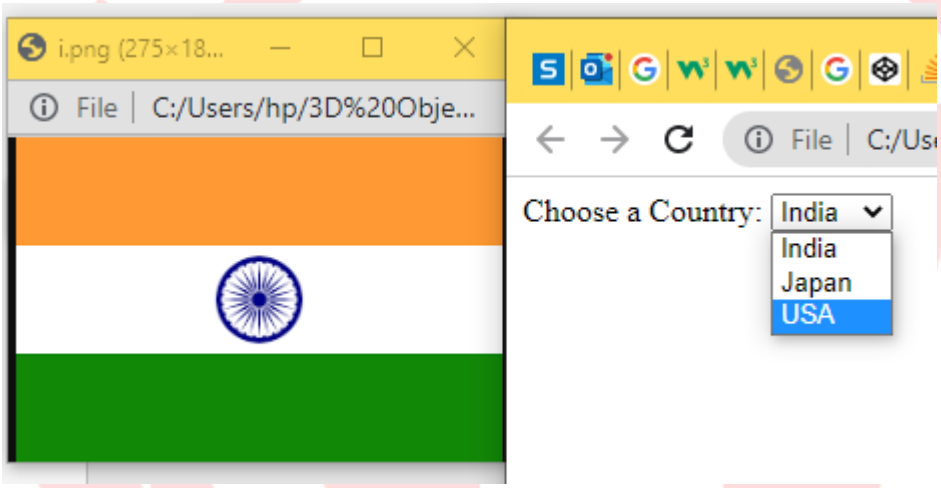
23. Write a javascript to create option list containing list of images and then display images in new window as per selection.

Code:

```
<html>
<body>
        <label for="contries">Choose a Country:</label>
        <select name="contries" id="contries" class="select"
onchange="changecontry()">
            <option value="india">India</option>
            <option value="japan">Japan</option>
            <option value="usa">USA</option>
        </select>

<script>
changecontry= () => {
   var contries = document.getElementById("contries").value;
   switch(contries)
{
        case "india":
        var myWindow=window.open("i.png","MsgWindow",
"width=200,height=100");
            break;
        case "japan":
        var myWindow=window.open("j.png","MsgWindow",
"width=200,height=100");
            break;
        case "usa":
             var myWindow=window.open("u.png","MsgWindow",
"width=200,height=100");
            break;
        default:
```

```
        /*document.getElementById("city-name").innerHTML = "Nill";
        document.getElementById("city-description").innerHTML = "Nill";
        document.getElementById("city-image").src = "";*/
        break;
    }
}</script>
</body>
</html>
```

Output:



24. Write a javascript function to generate Fibonacci series till user defined limit.

```
<script>
function fibonacci(num)
{
    var x = 0;
  var y = 1;
  var z;
  var i = 0;
document.write(x);
document.write("<br>"+y);
  for (i = 2; i < num; i++)
```

```
  {
    z = x + y;
    x = y;
    y = z;
document.write("<br>"+y);
  }
}
var num = parseInt(prompt('Enter the number of terms: '));
answer = fibonacci(num);
</script>
```

Output: