Javascript

Unit 1-

From Blackbook by Kogent Solutions

Javascript

- Javascript is one of the most simple, versatile and effective languages used to extend functionality in websites.

Advantages

- Javascript is executed on the client side: This means that the code is executed on the user's processor instead of the web server (saving bandwidth and strain on the web server)
- easy language
- fast to the end user

Javascript

Disadvantages

- Security Issues: JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system.
- JavaScript rendering varies: Different layout engines may render it differently resulting in inconsistency in terms of functionality and interface.

Objects in javascript

1.String: create/manipulate string enclosed in quotes

Methods:bold,charAt(),concat(),indexOf(),toUpperCase(), toLowerCase(),replace(),match(),search(),substring()

2.array: create /manipulate series of values rep by a single name

Methods:concat(),join(),reverse(),sort(),valueOf()

3.Date: create /manipulate dates and times

Methods:Date(),getDate(),getDay(),getMonth(),getYear()

String methods

- All string methods return a new value. They do not change the original variable.

String methods

- **var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";**
  var sln = txt.length;

- **var str = "Please locate where 'locate' occurs!";**
  var pos = str.indexOf("locate");

- **var str = "Apple, Banana , Kiwi";**
  var res = str.substring(7, 13);

- **str = "Please visit Microsoft!";**
  var n = str.replace("Microsoft", "Schools");

String methods

- **var text1 = "Hello World!"; // String**
  var text2 = text1.toUpperCase(); // text2 is text1 converted to upper
- **var str = "HELLO WORLD";**
  str.charAt(0); // returns H
- var str = "Hello World!";

  var result = str.bold();

  document.getElementById("demo").innerHTML = result;

String methods

```
var str = "The rain in SPAIN stays mainly in the plain";
 var res = str.match(/ain/g);
 document.getElementById("demo").innerHTML = res;
```

O/P: ain,ain,ain

String methods

```
<p id="demo"></p>
<script>
var text1 = "Hello";
var text2 = "World!";
var text3 = text1.concat(" ",text2);
document.getElementById("demo").innerHTML = text3;
var str = "Visit Us!";
var n = str.search(“Us");//case sensitive search occurs else -1
document.getElementById("demo").innerHTML = n; </script>
```

Declaring Arrays

Var bk=new Array(3);

or

Var bk=new Array();

or

Var bk=new Array(“brief”,”cheese”,”call”);

Declaring Arrays

```
Var bkPrices=new Array(3);
Var bknames=new Array(“Thesis”,”Tinkle”,”Chandamama”);
bkPrices[0]=50;
bkPrices[1]=100;
bkPrices[2]=150;
Document.write(bkPrices.join())//50,100,150
for (i = 0; i < bkPrices.length; i++) {
   document.write(bknames[i] + “cost Rs. “ + bkPrices [i] );
}
```

Array methods

1. length()
2. Reverse()
3. Sort()
4. Concat()-arrayname.concat(“value”);
5. Join()
6. valueOf(): converts array values into single value

Eg. var fruits = ["Banana", "Orange", "Apple", "Mango"];

document.getElementById("demo").innerHTML = fruits.valueOf();

O/p : Banana,Orange,Apple,Mango

Arrays

```
var cars = ["Saab", "Volvo", "BMW"];
var person = ["John", "Doe", 46];
document.getElementById("demo").innerHTML = cars[0];
var x = cars.length;   // The length property returns the number of
  elements
var y = cars.sort();   // The sort() method sorts arrays ascending order
```

Arrays

```
var fruits, text, fLen, i;
//display array
fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;
text = "<ul>";
for (i = 0; i < fLen; i++) {
    text += "<li>" + fruits[i] + "</li>";
}
```

Arrays

```
<p>Click the button to sort the array in ascending order.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
var points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo").innerHTML = points;
function myFunction() {
    points.sort(function(a, b){return a - b});//b-a descending
    document.getElementById("demo").innerHTML = points;
}
</script>
```

Arrays

```
<p>The sort() method sorts an array alphabetically.</p>
<p>The reverse() method reverts the elements.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;
function myFunction() {
    fruits.sort();//ascending order
    fruits.reverse();reverses array
    document.getElementById("demo").innerHTML = fruits;
}
</script>
```

Date object

var d = new Date(*year*, *month*, *day*, *hours*, *minutes*, *seconds*, *milliseconds*);

The getDate() method returns the day of a date as a number (1-31)

The getDay() method returns the weekday of a date as a number (0-6)//Sunday 0

The getFullYear() method returns the full year of a date//2020

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Date();
</script>
```

```
var d = new Date();

var months =
["January","February","March","April","May","June","July","August","September","Octob
er","November","December"];

document.write(d.getMonth());

document.getElementById("demo").innerHTML = months[d.getMonth()];//d.getMonth()
returns 9
```

Some more Javascript methods:

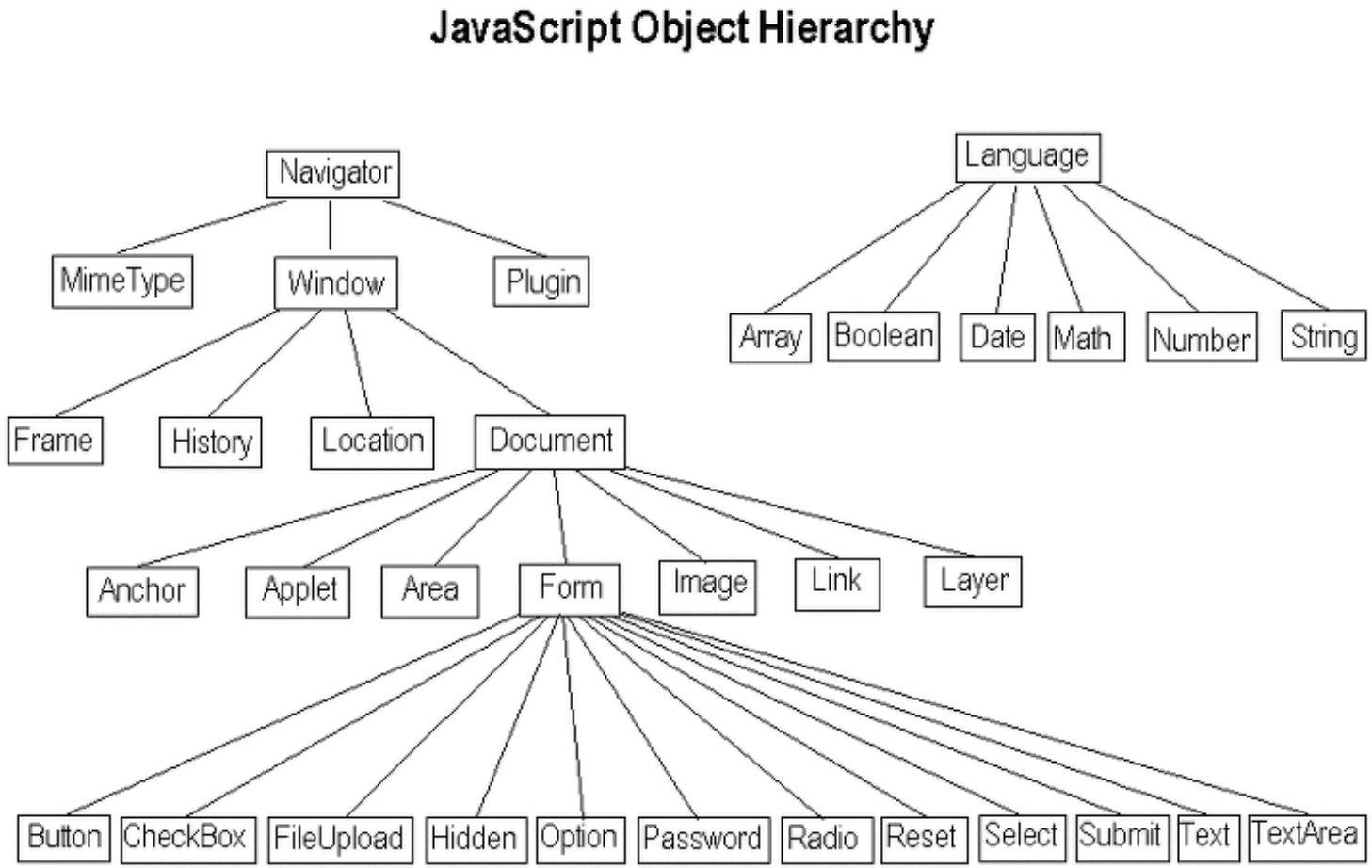
1.isFinite-returns true if value is finite
2.eval()-evaluates expression.Eg.

```
  var x = 10;
  var y = 20;
  var a = eval("x * y") + "<br>";
```

3.Escape()-encodes special characters except *,@,-,_,+,.,/
4.Unescape()-decode strings that are encoded

```
var str="@.Need Help? Visit!";
var str_esc=escape(str);
document.write(str_esc + "<br>")// @.Need%20Help%3F%20Visit%21
document.write(unescape(str_esc))//@.Need Help? Visit!
```

Object Heirarchy:



DOM objects

- **Window Object:** Window Object is at always at top of hierarchy. It's the web browser window —alert(),close(),confirm(),focus(),prompt(), print(),open()
- **Document object:** When HTML document is loaded into a window, it becomes a document object.It has Lastmodifiesd,URL and title along with close(),getelementByID(),open(),write(),writeln(),
- **Navigato**r: allows access info about browsers:- appname,appversion,cookieenabled,platform-javaenabled()
- **Screen**-screen info like height,width,color
- **Histor**y- urls opened IN PAST–back,forward,go()
- **Location**-current url opened –hostname,href,pathname,port,protocol---assign(),reload(),replace()

Basics

- **document.getElementById("demo")**.style.fontSize = "35px";
- document.getElementById("demo").innerHTML = "Hello Dolly.";
- var x = 5; // Declare x, give it the value of 5
- document.getElementById("myH").innerHTML = "My First Page";
  document.getElementById("myP").innerHTML = "My first para.";

Basics

```
<script type="text/javascript">
<!--
// This is a single line JavaScript comment
/*This is a multiple line JavaScript comment
document.write("I have comments in my JavaScript code!");
//document.write("You can't see this!");
*/
//-->
</script>
```

Variables

- Data stored in variables
- Name should be unique
- Should contain letters,numbers,underscores
- Must begin with uppercase or lowercase
- Name cannot be enclosed by double or single quotes
- Name cannot be a keyword

Jump statements

- break
- continue

Operators:Examples

Arithmetic: var1=5;++var1=6;var1++=5(similary for --)

Assignment : suppose var1 and 2=5 ;var1+=var2//10(same for others)

Comparison : ==(returns true or false)

?(ternary): result=var1<var2?var1:var2

Logical operators : &&,||,!

Operator precedence:Highest to lowest

!,-,++,--

*,/,%

+,-

<,<=,>,>=

==,!=

&&,||,?

=,+=,-=,*=,/=,%=

Addition/Concat Examples

Var x=5+5; //answer is 10

txt1=“hello”;

txt2=“dear”;

Txt3=txt1 + “ ” + txt2; //output is hello dear

Function

- function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
  }

- var x = toCelsius(77);
  var text = "The temperature is " + x + " Celsius";

Conditional statements

1.IF:

```
if (hour < 18) {
      greeting = "Good day";
  }
```

2.ELSE:

```
if (hour < 18) {
      greeting = "Good day";
  } else {
      greeting = "Good evening";
  }
```

Conditional statements

3.ELSE IF:

```
if (time < 10) {
      greeting = "Good morning";
  } else if (time < 20) {
      greeting = "Good day";
  } else {
      greeting = "Good evening";
  }
```

4.SWITCH:

```
switch (new Date().getDay()) {
        case 0:
           day = "Sunday";
           break;
        case 1:
           day = "Monday";
           break;
        case 2:
           day = "Tuesday";
           break;
        case 3:
           day = "Wednesday";
           break;
        case 4:
           day = "Thursday";
           break;
        case 5:
           day = "Friday";
           break;
        case 6:
           day = "Saturday";
           break;
        default:
            day=invalid day”;
            break;
}
```

Loops:For,While,DoWhile

- for (i = 0; i < cars.length; i++) {
  text += cars[i] + "
";
  }

- while (i < 10) {
  text += "The number is " + i;
  i++;
  }
- do {
  text += "The number is " + i;
  i++;
  }
  while (i < 10);

Conversion

There are 3 JavaScript methods that can be used to convert variables to numbers:

- The Number("10") method
- The parseInt() method
- The parseFloat() method

Pop up boxes

1. Alert box
2. Confirm box
3. Prompt box

Alert Box

- An alert box is often used if you want to make sure information comes through to the user.
- When an alert box pops up, the user will have to click "OK" to proceed.
- alert("I am an alert box!");

Confirm Box

- A confirm box is often used if you want the user to verify or accept something.
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
- If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.
- ```
  if (confirm("Press a button!")) {
      txt = "You pressed OK!";
  } else {
      txt = "You pressed Cancel!";
  }
  ```
  ```

Prompt Box

- A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.
- var person = prompt("Please enter your name", "Harry Potter");

```
if (person == null || person == "") {
    txt = "User cancelled the prompt.";
} else {
    txt = "Hello " + person + "! How are you today?";
}
```

External JavaScript:

```
<!DOCTYPE HTML>
<head>
<title>Add two numbers</title>
<script type="text/javascript" src="script.js" >
</head>
<body>
Enter the first number: <input type="text" id="txt1" /><br />
Enter the second number: <input type="text" id="txt2" /><br />
<input type="button" onclick="call()" value="Add"/>
</body>
</html>
```

Create a javascript file(script.js) with the below content.

```
function call(){
var q=parseInt(document.getElementById("txt1").value);
var w=parseInt(document.getElementById("txt2").value);
var result=q+w;
   if(isNaN(q)||isNaN(w)){
         alert("please enter a number");
     }
     else
        {
         var result=q+w;
         alert("The sum is " +result);
       }
}
```

External JavaScript Advantages

1. It separates HTML and code
2. It makes HTML and JavaScript easier to read and maintain
3. Cached JavaScript files can speed up page loads

HTML Events

- An HTML event can be something the browser does, or something a user does.

- Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

- Often, when events happen, you may want to do something.

- JavaScript lets you execute code when events are detected.

HTML Events

1. Onchange:

```
<img src="/images/mars-rover-no-laser.png"
id="image"
onmouseout="changeimage()">

<script>
function changeimage()
{
document.getElementById("image").src="/images/mars-rover-laser.png";
}
</script>
```

HTML Events

2.Onclick:

```
<img id="image" onclick="changeimage()" src="/images/Rock.png">

<script>
function changeimage()
{
document.getElementById("image").src= "/images/nature.png";
}
</script>
```

HTML Events

3.onmouseover

```
<img src="/images/rocket-before-takeoff.png"
id="image"
onmouseover="changeimage()">


<script>
function changeimage()
{
document.getElementById("image").src="/images/rocket-takeoff.png";
}
</script>
```

HTML Events

4.onmouseout

Eg1: <a href="http://www.learningaboutelectronics.com" onmouseout="alert(''You scrolled over the link and then moved the cursor away')"> Scroll over this link and then move the cursor away

Eg2: <img src="/images/mars-rover-no-laser.png"
alt="Mars exploration"
id="image2"
onmouseover="changeimage()"
onmouseout="changeimageback()">

function changeimageback()
{document.getElementById("image2").src="/images/mars-rover-no-laser.png";}

HTML Events

4.onkeydown

Enter Your Name: <input type="text" onkeydown="alert('You Pressed a Key');"/>

HTML Events

5.onload :

<body onload="alert ('Welcome to this site');">

6.onfocus event

```
<script>
function highlightyellow(x)
{
x.style.background="yellow";
}
</script>

Enter your name: <input type="text" onfocus="highlightyellow(this)">
```

7.onblur Event

- **HTML Code**
- The HTML code to create the above text box is:

  Enter Your Email: <input type="text" id="textbox2" onblur="turnblue(this)"/>

- **Javascript Code**
- The Javascript code to change the textbox to blue when clicked away from is:

```
<script>
function turnblue(x)
{
x.style.background="blue";
}
</script>
```

8.ondblclick Event

-occurs when user double clicks element

9.onerror Event

-occurs when error arises on page or image load

10.onkeypress/11.onkeyup Event

-occurs when user presses key for a while /releases key

12.onmouseup/13.onmousedown Event

-occurs when user releases mouse button/ presses mouse button

14.onselect Event

-occurs when user selects text or textarea on form

Write javascript to Validate Email id :

Solution:

```
<input type='text' id='txtEmail'/>
<input type='submit' name='submit' onclick='checkEmail();'/>
```

Validate Email id : Solution using string methods

```
function checkEmail()  {
var email = document.getElementById('txtEmail').value;
    // Get email parts
    var emailParts = email.split('@');
    // There must be exactly 2 parts
    if(emailParts.length !== 2) {
        alert("Wrong number of @ signs");
        return false;
    }
```

Validate Email id : Solution using string methods

```
// Name the parts
   var emailName = emailParts[0];
   var emailDomain = emailParts[1];
   // === Validate the parts === \\
   // emailName must only include valid chars
   for(var i = 0; i < emailName.length; i += 1) {
      if(validChars.indexOf(emailName.charAt(i)) < 0 ) {
         alert("Invalid character in name section");
         return false;
      }
   }
```

Solution using string methods:

```
// Must be at least one char before @ and 3 chars after
  if(emailName.length < 1 || emailDomain.length < 3) {
     alert("Wrong number of characters before or after @ sign");
     return false;
  }
```

Solution using string methods:

```
   // Define valid chars
var validChars =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','.',

'0','1','2','3','4','5','6','7','8','9','_','-'];
 // emailDomain must only include valid chars
   for(var j = 0; j < emailDomain.length; j += 1) {
      if(validChars.indexOf(emailDomain.charAt(j)) < 0) {
         alert("Invalid character in domain section");
         return false;
      }
   }
```

Solution using string methods:

```
    // Domain must include but not start with .
    if(emailDomain.indexOf('.') <= 0) {
        alert("Domain must include but not start with .");
        return false;
    }
    // Domain's last . should be 2 chars or more from the end
    var emailDomainParts = emailDomain.split('.');
    if(emailDomainParts[emailDomainParts.length - 1].length < 2) {
        alert("Domain's last . should be 2 chars or more from the end");
        return false;
    }
 alert("Email address seems valid");
 return true;
}
```

Semester questions

- Explain different pop up boxes?(6 marks)
- Design calculator using javascript?(6 marks)
- Explain with example how javascript code gets executed?(4 marks)
- Explain the working and the main components of search engine?

Explain with example how javascript code gets executed?(4 marks)

```
<script>
Function welcome()
{
document.write("Hello, World!");
}
</script>
```

The browser fetches a page which might have embedded JavaScript as we saw above, or refers to a separate JavaScript file in which case that file is also fetched by the browser

Next up, depending upon what you want your JavaScript code to do and how you structured it, it will either execute as soon as the file loads or wait for a triggering event (like a click, or load).

<input type=“submit” onclick='Javascript:welcome();'/>

Finally, when it’s all good, the code is executed line by line.

JavaScript is an interpreted language, which means you don’t need to compile the code into another form to execute it. All of the changes you make, instantly take effect.