

Web designing technologies
(Javascript-DHTML)

T.Y.B.Sc.(Computer Science)

Semester – IV

Paper : Internet Programming – II

Mr. Abhijeet D. Mankar

Department of Computer Science

Tuljaram Chaturchand College,

Baramati.

JAVASCRIPT

Facts about JavaScript

- Developed by **Brendan Eich**
- First supported by Netscape Navigator Browser in early 1996

Difference between Java and JavaScript

Java	JavaScript
Object oriented programming language	Object oriented scripting language
Creates applications that run in a virtual machine or browser	JavaScript code is run on a browser only

JavaScript is cross platform, object oriented scripting language used in web pages to perform tasks such as -

- Validation of data
- Processing of numbers and modification of form contents

Features of JavaScript

- **Only scripting language supported by all Web Browsers**
- Can be embedded directly in HTML pages or could be used from a separate file
- When included in an HTML web page, the script is interpreted and executed by the Web Browser along with the HTML code
- Case sensitive (e.g. variable 'NUM' \neq 'num')

Applications of JavaScript

- A universal programming tool for HTML designers :
Web page designers having very little knowledge of programming can write short codes of JavaScript, because of the simpler syntax.
- JavaScript can put dynamic text into an HTML page :
e.g. `document.write("<h1>"+name+"</h1>")`
- JavaScript can react to events.
- JavaScript can read and write HTML elements.
- JavaScript can be used to validate data.

JavaScript can be used for

Client Side Scripting

Validation of form input before sending the data to the server.

Server Side Scripting

The server executes the script and the output generated by the server as a result of this execution is downloaded by the client.

Variables

A variable name

- Always begins with letter
- Is without any blank space or special symbol except underscore(_)
- Is upto 255 characters long
- Cannot be reserved keyword
- Is case sensitive

Example :

- 1) `var num;` //variable 'num' is declared explicitly
`num=11;` //variable 'num' is assigned value 11
- 2) `var num=11;` // variable 'num' is declared explicitly &
// assigned value 11
- 3) `num=11;` // variable 'num' is declared implicitly &
// assigned value 11

Data Types

Simple(primitive) data types

➤ Number

0, -20, 1000, 2 : integers

1.11, 1.0e2 : floating-point numbers

NaN : It is the value returned by JavaScript when the program tries to treat something that is not a number as a number.

➤ String

Sequence of valid characters within a given character set.

Some escape characters are as follows:

\0 : Null character

\b : Backspace

\t : Tab

\n : New line

\" : Double quote

\' : Single quote

\\ : The backslash

➤ Boolean

Two Boolean values : **true** and **false**

Data Types

Advanced Data Types

➤ **Array**

Arrays are written with square brackets or round brackets.

e.g. `var cars = ["ABC", "LMN", "XYZ"];`

- OR -

`var cars=new Array("ABC", "LMN", "XYZ");`

➤ **Object**

Objects are written with curly braces.

e.g. `var person = {rollno:1, name:"abc", age:25}`

Literals

These are fixed values that you literally provide.

➤ Integers

A decimal integer literal : sequence of digits without leading 0

An octal integer literal : leading 0 on an integer literal

A hexadecimal integer literal : leading 0x or 0X

➤ Floating Point Literals

➤ Boolean Literals

Two Boolean values : `true` and `false`

➤ String Literals

Zero or more characters enclosed in double(“) or single(‘) quotes

Operators

Unary Operators

Operator	Meaning
++x	Pre-increment
x++	Post-increment
--x	Pre-decrement
x--	Post-decrement

Binary Operators

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
=	Assignment

Operators

Composite Assignment Operators

Operator	Meaning
<code>+=</code>	Composite Assignment Addition
<code>-=</code>	Composite Assignment Subtraction
<code>*=</code>	Composite Assignment Multiplication
<code>/=</code>	Composite Assignment Division
<code>%=</code>	Composite Assignment Remainder

Operators

Comparison Operators

Operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not Equal To

Logical Operators

Operator	Meaning	Example
AND	&&	Expr1 && Expr2
OR		Expr1 Expr2
NOT	!	!Expr1

Comments

There are different styles of comments

- Single line comment : `//`
- Multiline text comment : `/*...*/`
- Multiline : `<!--`
statements
`-->`

Insertion of JavaScript

➤ JavaScript in the <head> section

Scripts which are to be executed only when called or when an event is triggered.

➤ JavaScript in the <body> section

Scripts which are to be executed while the page loads.

➤ External JavaScript

Script can be written in an external file, and saved with a .js file extension. The external script must not contain the <script> tag.

Creating user defined function

➤ In order to create a function, we have to define

✓ Function name

✓ Function Parameters

➤ Syntax

```
function fname(arg1,arg2,...)
{
    statements
}
```

➤ If a function is placed in the <head> section of the web page, all code in the function gets loaded in the computer's memory before the function is called.

JavaScript Built-in Functions

➤ `eval()`

Accepts string as an argument which can be statement, expression or sequence of statements.

➤ `parseInt()`

Returns only the first number in the string. If character cannot be converted to a number it returns NaN.

➤ `parseFloat()`

Returns only the first number in the string.

➤ `confirm()`

When confirm box pops up, user has to click either “OK” or “Cancel” to proceed.

➤ `prompt()`

It displays the argument of the function in the message along with two buttons labeled “OK” and “Cancel”.

➤ `alert()`

When an alert box pops up, user will have to click “OK” to proceed.

Statements in JavaScript

➤ If statement

Syntax :

```
    if(condition)
    {
        statement1;
        statement2;
    }
    else
    {
        statement3;
        statement4;
    }
    statement5;
```

If condition evaluates to TRUE, then statement1 and statement2 are executed, after which control will be transferred to the execution of statement5.

Statements in JavaScript

➤ Conditional Expression Operator

Syntax :

(condition?expression1:expression2)

↑
if

↑
true

↑
else

If condition evaluates to TRUE, then the whole operator is assigned the value of expression1 else it is assigned the value of expression2.

➤ The switch statement

Syntax :

```
switch(expression)
{
    case constant1:statements; break;
    case constant2:statements; break;
    ...
    case constantN:statements; break;
    default:statements;
}
```

Iteration (looping) Statements

➤ while loop

Syntax :

```
while(condition)
{
    statement1;
    statement2;
}
statement3;
```

If condition evaluates to TRUE, then the body of the loop (statement1 and statement2) will be executed, Otherwise statement3 will be executed.

➤ do...while loop

Syntax :

```
do
{
    statement1;
    statement2;
} while(condition)
statement3;
```

Iteration (looping) Statements

➤ for loop

Syntax :

for(initialization; condition; increment or decrement)

➤ break and continue

The break keyword can be used to transfer control outside the loop.
(to come out of the loop)

The continue keyword can be used to jump to the next iteration of the loop.

Some programs

1. Javascript program to display greeting message (onload event)

```
<html>
<head><title>Greetings!</title>
<script type="text/javascript">
function display_greeting()
{
var d=new Date();
var date=d.getDate();
var mon=d.getMonth();           //0-11
var year=d.getFullYear();
var day=d.getDay();
var hour=d.getHours();
var mins=d.getMinutes();
var sec=d.getSeconds();
var greet="";
mon+=1;
greet+=" "+date+"/"+mon+"/"+year+" ";
switch(day)
{
    case 0:    greet+="Sunday ";
               break;
    case 1:    greet+="Monday ";
               break;
    case 2:    greet+="Tuesday ";
               break;
    case 3:    greet+="Wednesday ";
               break;
    case 4:    greet+="Thursday ";
               break;
    case 5:    greet+="Friday ";
               break;
    case 6:    greet+="Saturday ";
               break;
}
if(hour<12)
{
    greet+="<b>Good Morning!</b>"+"it's"+hour+":"+mins+":"+sec;
}
else if(hour>=12 && hour<16)
{
    greet+="<b>Good Afternoon!</b>"+"it's"+hour+":"+mins+":"+sec;
}
else if(hour>=16 && hour<20)
{
    greet+="<b>Good Evening!</b>"+"it's"+hour+":"+mins+":"+sec;
}
else
{
    greet+="<b>Good Night!</b>"+"it's"+hour+":"+mins+":"+sec;
}
document.write(greet);
}
</script>
</head>
<body onload="display_greeting()">
</body>
</html>
```


2. Javascript program to format the text (onclick event)

```
<html>
<head>
<script type="text/javascript">
function do_format(msg) {
    document.write("The original string: " + msg);
    document.write("<p>Bold: " + msg.bold() + "</p>");
    document.write("<p>Italic: " + msg.italics() + "</p>");
    document.write("<p>Underline: <u>" + msg + "</u></p>");
    document.write("<p>Upper Case: " + msg.toUpperCase() + "</p>");
    document.write("<p>Lower Case: " + msg.toLowerCase() + "</p>");
    document.write("<p>Strikethrough: " + msg.strike() + "</p>");
    document.write("<p>Link: " + msg.link("http://www.abc.com") + "</p>");
    document.write("<p>Fontcolor: " + msg.fontcolor("red") + "</p>");
}
</script>
</head>
<body>
<form>
<input type="text" id="message">
<input type="button" value="Click" onclick="do_format(message.value)">
</form>
</body>
</html>
```

3. Javascript program to display user defined message in specified colour

```
<html>
<head>
<script type="text/javascript">
function disp_msg(){
    var msg=document.getElementById("tmsg").value;
    var color=document.getElementById("clr").value;
    document.getElementById("content").innerHTML=msg.fontcolor(color);
}
</script>
<body>
<form>
<pre>
    Enter Your Message <input type="text" id="tmsg">
    Choose Colour      <input type="text" id="clr">
    <div id="content">Display Your Message Here</div>
    <input type="button" value="Click" onclick="disp_msg()">

</pre>
</form>
</body>
</head>
</html>
```

4. Javascript program to demonstrate onmouseover and onmouseout events

```
<html>
<head>
<script type="text/javascript">
    // Called when mouse pointer is moved onto an area
        function MouseIn() {
            document.getElementById("myarea").style.border = "2px solid blue";
        }
    // Called when mouse pointer is taken away from an area
    function MouseOut() {
        document.getElementById("myarea").style.border = "";
    }
</script>
</head>
<body>
    <div style="background-color:cyan; width:200px" id="myarea" onmouseover="MouseIn()"
        onmouseout="MouseOut()"> Move mouse pointer into and out of this.
    </div>
</body>
</html>
```

5. Javascript program to demonstrate onmouseup and onmousedown events

```
<html>
<head>
<script type="text/javascript">
//Called when a mouse button is pressed
function mouseDown() {
    document.getElementById("para").style.color = "red";
}
//Called when a mouse button is released
function mouseUp() {
    document.getElementById("para").style.color = "blue";
}
</script>
</head>
<body>
<form>
<p id="para" onmousedown="mouseDown()" onmouseup="mouseUp()">
    Click the Mouse Button Here to see the change
</p>
</form>
</body>
</html>
```

6. Javascript program to demonstrate onsubmit event

```
<html>
<head>
<script type="text/javascript">
function chk_user(){
  var usr=document.getElementById("user").value;
  var pwd=document.getElementById("pass").value;
  if(usr=="" || pwd=="") {
    alert("User Name Or Password Cannot Be Empty!");
  }
  if((usr.length<6) || (usr.length>15)){
    alert("User Name : between 6 to 15");
  }
  if((pwd.length<8) || (pwd.length>20)){
    alert("Password : between 8 to 20");
  }
}
</script>
</head>
<body>
<form name="myform" onsubmit="chk_user()">
<pre>
Enter User Name <input type="text" id="user">
Enter Password <input type="password" id="pass">
<input type="submit" value="SignIn">
</pre>
</form>
</body>
</html>
```

7. Javascript program to demonstrate onkeydown, onkeypress, onkeyup events

```
<html>
<head>
<script type="text/JavaScript">
function keyDown() {
alert("Key pressed");
}
function keyUp() {
alert("Key released");
}
function keyPress(){
alert("Key pressed and released");
}
</script>
</head>
<body>
<pre>
    Type in this textbox for keyDown <input type="text" onkeydown="keyDown()">
    Type in this textbox for keyUp  <input type="text" onkeyup="keyUp()">
    Type in this textbox for keyPress <input type="text" onkeypress="keyPress()">
</pre>
</body>
</html>
```

Note : onkeydown event is fired for all keys regardless of whether the key produce a character value.