(WD)

BSCIT

web designing

(today's date)

JavaScript:

-HTML Creates-Static Pages..i.e Pages in which their contents cannot be changed. So that you need a Programming/Scripting language.

Difference between Programming language and Scripting language:

- Progg. Langg e.g, C++, Java
  - Must be compiled.
  - Must be translated into machine code before execution
  - Complex to learn
- Scripting Langg e.g, Javascript, VBScript
  - Need to be interpreted not to be compiled
  - Browser executes each line of script as it comes to it.

easy to learn

Advantages /Disadvantages of Scripting Language:

- Advantages :
  - Has Simple syntax.. so easy to learn.
  - Can perform tasks with minimum number of commands.
  - Writing/Changing scripts are very simple and the change will be start acting as soon as the page will be reloaded in browser.
- Disadvantage :
  - Can't execute really quickly. So complex tasks like handling Graphics is a bit complicated.

Scripting: The script executed by the web server & procced by the server is called server side script

1) Server-Side Scripting → SSI, PHP, coldfusion, Asp, Asp.net
   - Scripts are executed on Server → It is not visible to user
   - As data is sent to Server and output is being received by client. Response time may be slow. → It is browser independent
   - Can read and write Files. → It takes more time to execute

2)Client-Side Scripting Javascript, VBScript → It is browser dependent
   - Scripts are executed on the Client (i.e. Browser) → It takes less time to execute
   - Cannot read files. They can read information from a form and can manipulate them but cannot store data on Web Server. → The scripting code is visible to user so it is not secure

Script executed only by the browser without contect in the server is called client-side script

JavaScript:-

-JavaScript is a scripting language that enables web developers/designers to build more functional and interactive websites.

- First scripting language introduced by Netscape Communication Corporation (makers of Netscape Navigator Web Browser) in 1995.
- Was originally called LiveScript and later was renamed JavaScript to indicate its relationship with Java.
- JavaScript is supported by all internet browsers and was standardized by the European Computer Manufacturers Association (ECMA).

Common uses of JavaScript include:

Javascript was designed to add interactivity to HTML pages.

- Alert messages
- Popup windows
- Dynamic dropdown menus
- Form validation
- Displaying date/time

→ Progs CGI → It entires executes on the server.

What You can do with Javascript ?

- Add scrolling / changing messages to the browser's status bar.
- Validate contents of a form and also make calculations.
- Display message to user.
- Animate images.
- Detect browser in use and can handle it.
- Detect installed plug-ins and notify user if it is required.
- Create clocks and calendars
- Create rotating banners using JPEG files
- Open new windows

JavaScript Framework:-

- Framework for JavaScript

```
<script language="Javascript">
    //      Your script goes here..
</script>
```

❖ Including Scripts in an HTML Document:

There are actually **four ways** to include script code in a document:

- Within a **<script>**tag.
- As a linked **.js file** indicated by the **src** attribute of a **<script>**tag.
- Within an event handler attribute such as **onclick.**
- Via the pseudo-URL **javascript:** syntax referenced by a link.

1)The **<script>**tag:

The Script element is used to section off any script included directly within a web page.

Attributes :

1)Language:-Type of Script to Execute
2)Type:-Text File Containing JavaScipt
3)Src:Script location

Example:

-<Script
Language="javascript"Type=Text/JavaScript"Src="url"></Script>

The <Script>tag can occur in either the head or the body elements numerous times.

Places of <script>Tag:

1)In the body
2)In the header
3)Event handler,html tag
4)External Document

Order in Which Scripts run:

1)First header Script runs.
2)Body Scripts.
3)Event Handlers.

Two places you can put your JavaScript in an HTML page: In either the head or body section.

1) Place JavaScript in the Body Section of the HTML page:

-In this, JavaScript is embedded with the Script tags somewhere in the body section of the HTML page. Scripts embedded in the body section are executed as part of the HTML document when the page loads.

Example:

```
<BODY>
<Script Language="JavaScript" Type=text/javascript">
document.write("This JavaScript is located in the body section")
</Script>
</BODY>
```

2)Placing JavaScript in the Head Section of the HTML Page:

→ Scripts used to be executed when an event is triggered, go in the head section.

In this,the JavaScript is embedded with the Script tags at the top of the HTML page within the <head></head>tags.Scripts embedded in the head section are not automatically when the page loads but can be excuted when called by other script in the page.

Example:

```
<head>
<title>Simple HTML Page</title>
<Script Language="Javascript"type="text/javascript">
alert("This JavaScript is located in the head section")
</Script>
</head>
```

3)Referencing JavaScript in an External .js File:

-In this SRC attribute is set to reference an external file that contains the JavaScript statements.

Example:

```
<Script Src="Filename.js"Language=:javascript"Type="text/javascript'>
</script>
```

Note:External JavaScript files can contain only JavaScript Statements.They cannot contain HTML tags.

4)Placing JavaScript in an HTML Tag:

-JavaScript can also be placed with the HTML tags as shown in the following example.In this case,the JavaScript onLoad=document.write("This JavaScript is located inside the BODY tag"); statement has been added to the HTML <BODY> tag.This particular Javascript statement tells the browser to write the enclosed text when the browser first loads the HTML page.

Example:

```
<Body onLoad = document.write ("This javascript
                is located inside the BODY tag")>

</Body>
```

* How to handle older browsar!
- Browsers that do not support javascript will display the script as page content. To prevent them from doing this, we may use the HTML comment tag:

```
<Script type="text/javascript">
<!--
        document.write("Hello World")
//→
```

The two forward slashes at the end of comment line (//) are a JavaScript comment symbol. This prevents the JavaScript compiler from compiling the line.

(1) Variables:- are used to store data.

Like other programming languages, JavaScript variables are a container that contains a value - a value that can be changed as required.
-A variable is a "container" for information you want to store. A variable's can change during the script.
Rules for declaring variable:

- Can include A-Z, a-z, 0-9, _
- No punctuation characters (eg comma, full stop, etc).
- The first character of a variable name cannot be a digit.
- JavaScript variables' names are case-sensitive. For example *firstName* and *FirstName* are two different variables.
- First Char of variable name should be either letter or "_"
- No Limit on length of a variable name but it must fit within one line
- E.g. total_no_of_flowers, LastAccNo, temp1, _var39, a ..... Etc.

not valid
100-nu
Rate-of-

❖ A variable cab be either local or global.

Local variable:

-A local variable is one that is explicitly declared using the var statement inside a function.

Global Variable:

-A global variable is any variable define outside of a function.

❖ Declaring JavaScript variables

First, you need to declare your variables. You do this using the **var** keyword. You can declare one variable at a time or more than one. You can also assign values to the variables at the time you declare them.

❖ Different methods of declaring JavaScript variables

```
// Declaring one javascript variable
var firstName;
// Declaring multiple javascript variables
var firstName, lastName;
//Declaring and assigning one javascript variable
var firstName = 'Homer';
// Declaring and assigning multiple javascript variables
var firstName = 'Homer', lastName = 'Simpson';
```

Using JavaScript variables:-

Example::

```
<script language="javascript" type="text/javascript" >
<!-- hide me
```

Datatype

(1) number → +ve, -ve & float
(2) String → Hello
(3) Boolean → true,

Complex datatype

(1) Array
(2) object

```
<Script type="text/javascript">
<!--
        document.write("Hello World")
//→
```

The two forward slashes at the end of comment line (//) are a JavaScript comment symbol. This prevents the JavaScript compiler from compiling the line.

(1) Variables:-

Like other programming languages, JavaScript variables are a container that contains a value - a value that can be changed as required.
-A variable is a "container" for information you want to store. A variable's can change during the script.
Rules for declaring variable:

- Can include A-Z, a-z, 0-9, _
- No punctuation characters (eg comma, full stop, etc).
- The first character of a variable name cannot be a digit.
- JavaScript variables' names are case-sensitive. For example *firstName* and *FirstName* are two different variables.
- First Char of variable name should be either letter or "_"
- No Limit on length of a variable name but it must fit within one line
- E.g. total_no_of_flowers, LastAccNo, temp1, _var39, a ..... Etc.

❖ A variable cab be either local or global.

Local variable:

-A local variable is one that is explicitly declared using the var statement inside a function.

Global Variable:

-A global variable is any variable define outside of a function.

❖ Declaring JavaScript variables

First, you need to declare your variables. You do this using the **var** keyword. You can declare one variable at a time or more than one. You can also assign values to the variables at the time you declare them.

❖ Different methods of declaring JavaScript variables

```
// Declaring one javascript variable
var firstName;
// Declaring multiple javascript variables
var firstName, lastName;
//Declaring and assigning one javascript variable
var firstName = 'Homer';
// Declaring and assigning multiple javascript variables
var firstName = 'Homer', lastName = 'Simpson';
```

Using JavaScript variables:-

Example::

```
<script language="javascript" type="text/javascript" >
<!-- hide me
```

```
var firstName = prompt("What's your first name?", "");
// end hide -->
<!-- hide me
document.write(firstName);
// end hide -->
</script>
```

The above example opens a JavaScript prompt, prompting the user for their first name. It will then write the name to the page (in practice, you would output the name somewhere between the <body></body> tags)

❖ Lifetime of Variables:-

When you declare a variable within a function, the variable can only be accessed within that function. When you exit the function, the variable is destroyed. These variables are called local variables. You can have local variables with the same name in different functions, because each is recognized only by the function in which it is declared.

If you declare a variable outside a function, all the functions on your page can access it. The lifetime of these variables starts when they are declared, and ends when the page is closed.

(2)Conditional Statement:

In JavaScript we have the following Conditional Statements:

- If statements:-Use this statement if you want to execute some code only if a specified is true.
- If..else statement:-Use this statement if you want to execute some code if the condition is true and another code if the condition is false
- If..else if..else statement:-use this statement if you want to select one of many blocks of code to be executed
- Switch statement:-Use this statement if you want to select one of many blocks of code to be executed.

If Statement:-

You should use the if statement if you want to execute some code only if a specified condition is true.

Syntax-

```
if (condition)
{
code to be executed if condition is true
}
```

Note that if is written in lowercase letters. Using uppercase letters (IF) will generate a JavaScript error!

Example 1

```
<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10
var d=new Date()
var time=d.getHours()

if (time<10)
```

Sun Mar 04 2007 05:33:59 GMT-0500

```
{
document.write("<b>Good morning</b>")
}
</script>
```

Example 2

```
<script type="text/javascript">
//Write "Lunch-time!" if the time is 11
var d=new Date()
var time=d.getHours()

if (time==11)
{
document.write("<b>Lunch-time!</b>")
}
</script>
```

Note: When **comparing** variables you must always use two equals signs next to each other (==)!

Notice that there is no ..else.. in this syntax. You just tell the code to execute some code **only if the specified condition is true**.

If...else Statement:-

If you want to execute some code if a condition is true and another code if the condition is not true, use the if....else statement.

Syntax

```
if (condition)
{
code to be executed if condition is true
}
else
{
code to be executed if condition is not true
}
```

Example

```
<script type="text/javascript">
//If the time is less than 10,
//you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.
var d = new Date()
var time = d.getHours()

if (time < 10)
{
document.write("Good morning!")
}
else
{
document.write("Good day!")
}
```

```
</script>
```

If...else if...else Statement

You should use the if....else if...else statement if you want to select one of many sets of lines to execute.

Syntax

```
if (condition1)
{
code to be executed if condition1 is true
}
else if (condition2)
{
code to be executed if condition2 is true
}
else
{
code to be executed if condition1 and
condition2 are not true
}
```

Example

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>")
}
else if (time>10 && time<16)
{
document.write("<b>Good day</b>")
}
else
{
document.write("<b>Hello World!</b>")
}
</script
```

The JavaScript Switch Statement

You should use the switch statement if you want to select one of many blocks of code to be executed.

Syntax

```
switch(n)
{
case 1:
  execute code block 1
  break
case 2:
  execute code block 2
```

```
 break
default:
 code to be executed if n is
 different from case 1 and 2
}
```

This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use **break** to prevent the code from running into the next case automatically.

Example

```
<script type="text/javascript">
//You will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.
var d=new Date()
theDay=d.getDay()
switch (theDay)
{
case 5:
 document.write("Finally Friday")
 break
case 6:
 document.write("Super Saturday")
 break
case 0:
 document.write("Sleepy Sunday")
 break
default:
 document.write("I'm looking forward to this weekend!")
}
</script>
```

Fri Mar 2 09:34:29 UTC+0530 2007

(3)JavaScript Operator:

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2.5

%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operators

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

String Operator

A string is most often text, for example "Hello World!". To stick two or more string variables together, use the + operator.

```
txt1="What a very"
txt2="nice day!"
txt3=txt1+txt2
```

The variable txt3 now contains "What a verynice day!".
To add a space between two string variables, insert a space into the expression, OR in one of the strings.

```
txt1="What a very"
txt2="nice day!"
txt3=txt1+" "+txt2
or
txt1="What a very "
txt2="nice day!"
txt3=txt1+txt2
```

The variable txt3 now contains "What a very nice day!".

Conditional Operator

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax

```
variablename=(condition)?value1:value2
```

Example

```
greeting=(visitor=="PRES")?"Dear President ":"Dear "
```

If the variable visitor is equal to PRES, then put the string "Dear President " in the variable named greeting. If the variable visitor is not equal to PRES, then put the string "Dear " into the variable named greeting.

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.

Syntax:

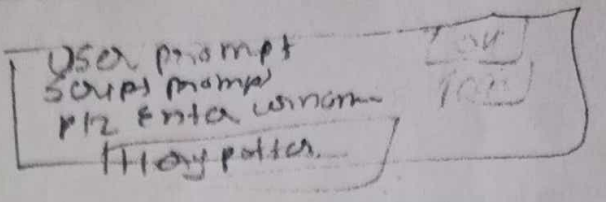
```
alert("some text")
```

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.

Syntax:

```
Confirm("sometext")
```



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.

Syntax:

```
prompt("sometext","defaultvalue")
```

JavaScript Functions

To keep the browser from executing a script as soon as the page is loaded, you can write your script as a function.

A function contains some code that will be executed only by an event or by a call to that function.

You may call a function from anywhere within the page (or even from other pages if the function is embedded in an external .js file).

Functions are defined at the beginning of a page, in the <head> section.

Example

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!")
}
</script>
</head>
<body>
<form>
<input type="button" value="Click me!"
onclick="displaymessage()" >
</form>
</body>
</html>
```

If the line: alert("Hello world!!"), in the example above had not been written within a function, it would have been executed as soon as the line was loaded. Now, the script is not executed before the user hits the button. We have added an onClick event to the button that will execute the function displaymessage() when the button is clicked.

How to Define a Function

The syntax for creating a function is:

```
function functionname(var1,var2,...,varX)
{
some code
}
```

var1, var2, etc are variables or values passed into the function. The { and the } defines the start and end of the function.

Note: A function with no parameters must include the parentheses () after the function name:

```
function functionname()
{
some code
}
```

Note: Do not forget about the importance of capitals in JavaScript! The word function must be written in lowercase letters, otherwise a JavaScript error occurs! Also note that you must call a function with the exact same capitals as in the function name.

The return Statement

The return statement is used to specify the value that is returned from the function. So, functions that are going to return a value must use the return statement.

Example

The function below should return the product of two numbers (a and b):

```
function prod(a,b)
{
x=a*b
return x
}
```

When you call the function above, you must pass along two parameters:

```
product=prod(2,3)
```

The returned value from the prod() function is 6, and it will be stored in the variable called product.

JavaScript Loops

Very often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.

In JavaScript there are two different kind of loops:

for - loops through a block of code a specified number of times

while - loops through a block of code while a specified condition is true

---

The for Loop

The for loop is used when you know in advance how many times the script should run.

Syntax

```
for (var=startvalue;var<=endvalue;var=var+increment)
{
  code to be executed
}
```

Example

Explanation: The example below defines a loop that starts with i=0. The loop will continue to run as long as **i** is less than, or equal to 10. **i** will increase by 1 each time the loop runs.

Note: The increment parameter could also be negative, and the <= could be any comparing statement.

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

```
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
document.write("The number is " + i)
document.write("<br />")
}
</script>
</body>
</html>
```

O/P: The no is 0 ... 10

The while loop

The while loop is used when you want the loop to execute and continue executing while the specified condition is true.

```
while (var<=endvalue)
{
  code to be executed
}
```

Note: The <= could be any comparing statement.

Example

Explanation: The example below defines a loop that starts with i=0. The loop will continue to run as long as **i** is less than, or equal to 10. **i** will increase by 1 each time the loop runs.

```
<html>
<body>
<script type="text/javascript">
var i=0
while (i<=10)
{
document.write("The number is " + i)
document.write("<br />")
i=i+1
}
</script>
</body>
</html>
```

O/P The no is 0 ... 10

The do...while Loop

The do...while loop is a variant of the while loop. This loop will always execute a block of code ONCE, and then it will repeat the loop as long as the specified condition is true. This loop will always be executed at least once, even if the condition is false, because the code is executed before the condition is tested.

```
do
{
  code to be executed
}
while (var<=endvalue)
```

Example

```
<html>
```

```
<body>
<script type="text/javascript">
var i=0
do
{
document.write("The number is " + i)
document.write("<br />")
i=i+1
}
while (i<0)
</script>
</body>
</html>
```

Result

```
The number is 0
```

JavaScript break and continue Statements

There are two special statements that can be used inside loops: break and continue.

Break

The break command will break the loop and continue executing the code that follows after the loop (if any).

Example

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
if (i==3){break}
document.write("The number is " + i)
document.write("<br />")
}
</script>
</body>
</html>
```

O/P:

The no is 0
1
2

Continue

The continue command will break the current loop and continue with the next value.

Example

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
if (i==3){continue}
document.write("The number is " + i)
```

O/P The no is 0

```
document.write("<br />")
}
</script>
</body>
</html>
```

JavaScript For...In Statement

The for...in statement is used to loop (iterate) through the elements of an array or through the properties of an object.

The code in the body of the for ... in loop is executed once for each element/property.

Syntax

```
for (variable in object)
{
  code to be executed
}
```

The variable argument can be a named variable, an array element, or a property of an object.

Example

Using for...in to loop through an array:

```
<html>
<body>
<script type="text/javascript">
var x
var mycars = new Array()
mycars[0] = "Saab"
mycars[1] = "Volvo"
mycars[2] = "BMW"

for (x in mycars)
{
document.write(mycars[x] + "<br />")
}
</script>
</body>
</html>
```

JavaScript break and continue Statements

There are two special statements that can be used inside loops: break and continue.

Break

The break command will break the loop and continue executing the code that follows after the loop (if any).

Example

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
```

```
if (i==3){break}
document.write("The number is " + i)
document.write("<br />")
}
</script>
</body>
</html>
```

Continue

The continue command will break the current loop and continue with the next value.

Example

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
if (i==3){continue}
document.write("The number is " + i)
document.write("<br />")
}
</script>
</body>
</html>
```

JavaScript For...In Statement

The for...in statement is used to loop (iterate) through the elements of an array or through the properties of an object.

The code in the body of the for ... in loop is executed once for each element/property.

Syntax

```
for (variable in object)
{
  code to be executed
}
```

The variable argument can be a named variable, an array element, or a property of an object.

Example

Using for...in to loop through an array:

```
<html>
<body>
<script type="text/javascript">
var x
var mycars = new Array()
mycars[0] = "Saab"
mycars[1] = "Volvo"
mycars[2] = "BMW"

for (x in mycars)
{
document.write(mycars[x] + "<br />")
```

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

Events

By using JavaScript, we have the ability to create dynamic web pages. Events are actions that can be detected by JavaScript.
Every element on a web page has certain events which can trigger JavaScript functions. For example, we can use the onClick event of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags.
Examples of events:

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input box in an HTML form
- Submitting an HTML form
- A keystroke

Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!

Insert Special Characters

The backslash (\) is used to insert apostrophes, new lines, quotes, and other special characters into a text string.
Look at the following JavaScript code:

```
var txt="We are the so-called "Vikings" from the north."
document.write(txt)
```

In JavaScript, a string is started and stopped with either single or double quotes. This means that the string above will be chopped to: We are the so-called
To solve this problem, you must place a backslash (\) before each double quote in "Viking". This turns each double quote into a string literal:

```
var txt="We are the so-called \"Vikings\" from the north."
document.write(txt)
```

JavasScript will now output the proper text string: We are the so-called "Vikings" from the north.
Here is another example:

```
document.write ("You \& me are singing!")
```

The example above will produce the following output:

```
You & me are singing!
```

The table below lists other special characters that can be added to a text string with the backslash sign:

Code	Outputs
\'	single quote
\"	double quote

\&	ampersand
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

JavaScript is Case Sensitive

A function named "myfunction" is not the same as "myFunction" and a variable named "myVar" is not the same as "myvar".

JavaScript is case sensitive - therefore watch your capitalization closely when you create or call variables, objects and functions.

White Space

JavaScript ignores extra spaces. You can add white space to your script to make it more readable. The following lines are equivalent:

```
name="Hege"
name = "Hege"
```

Break up a Code Line

You can break up a code line **within a text string** with a backslash. The example below will be displayed properly:

```
document.write("Hello \
World!")
```

However, you cannot break up a code line like this:

```
document.write \
("Hello World!")
```

Comments

You can add comments to your script by using two slashes //:

```
//this is a comment
document.write("Hello World!")
```

or by using /* and */ (this creates a multi-line comment block):

```
/* This is a comment
block. It contains
several lines */
document.write("Hello World!")
```

Object Oriented Programming

JavaScript is an Object Oriented Programming (OOP) language. An OOP language allows you to define your own objects and make your own variable types.

However, creating your own objects will be explained later, in the Advanced JavaScript section. We will start by looking at the built-in JavaScript objects, and how they are used. The next pages will explain each built-in JavaScript object in detail.

Note that an object is just a special kind of data. An object has properties and methods.

Properties

Properties are the values associated with an object.

In the following example we are using the length property of the String object to return the number of characters in a string:

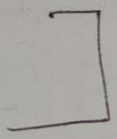
```
<script type="text/javascript">
var txt="Hello World!"
document.write(txt.length)
</script>
```

Methods

Methods are the actions that can be performed on objects.

In the following example we are using the toUpperCase() method of the String object to display a text in uppercase letters:

```
<script type="text/javascript">
var str="Hello world!"
document.write(str.toUpperCase())
</script>
```



```
                return false
        }
        if(IsValid())
        {
                return true
        }
        return false
}
function IsValid()
{
        var str=document.form1.email.value
        len=str.length
        for(i=1;i<(len-3);i++)
        {
                if(str.charAt(i)=="@")
                {
                        return true
                }
        }
        alert("You have entered invalid email address")
        return false
        }

function processForm()
{
        if(!Validate_FirstName(document.form1.fname.value))
        return false
        if(!Validate_LastName(document.form1.lname.value))
        return false
        if(!Validate_Email(document.form1.email.value))
        return false
        if(!IsValid())
        return false
        disp=open("","result")
        disp.document.write("<TITLE>Result Page</TITLE>" + "<PRE>")
        disp.document.write("<H2 ALIGN='CENTER'>" + "Thanks For Signing In" +
"</H2>" +"<HR>" +"<BR>")
        disp.document.write("First Name\t\t: " +document.form1.fname.value + "<BR>")
        disp.document.write("Last Name\t\t: " +document.form1.lname.value + "<BR>")
        disp.document.write("Email\t\t: " +document.form1.email.value + "<BR>")

        disp.document.write("Your Comments\t\t: " +document.form1.comment.value +
"<BR>")
        disp.document.write("</PRE>")
        if(disp.confirm("Is this information corrrect?"))
        disp.close()
```

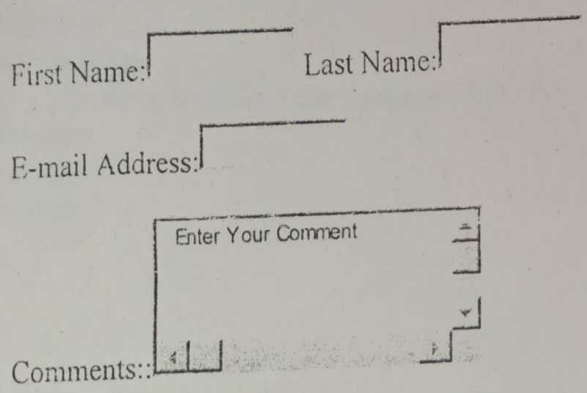
Upasana. S. Me

JavaScript-Validation Example:-

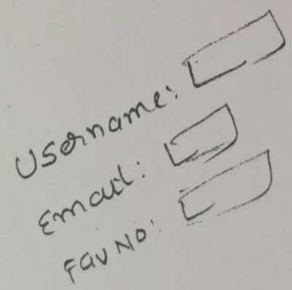
```
<HTML>
<head>
<title>Form Events</title>
<script LANGUAGE="Javascript">
function IsBlank(s) //check any field is empty or not.
{
        var len=s.length;
        for(i=0;i<len;i++)
        {
                if(s.charAt(i)!=" ")

                        {
                        return false
                        }

        }
        return true

}
function Validate_LastName()
{
        var str=document.form1.lname.value
        if(IsBlank(str))
        {
                alert("The Last Name field cannot be empty")
                return false
        }
        return true
}
function Validate_FirstName()
{
        var str=document.form1.fname.value
        if(IsBlank(str))
        {
                alert("The First Name field cannot be empty")
                return false
        }
        return true
}
function Validate_Email()
{
        var str=document.form1.email.value
        if(IsBlank(str))
        {
                alert("The email Name field cannot be empty")
```

```
}

</script>
</head>
<body>
<form name="form1">
<P>First Name:<input type="text" name=fname onChange="Validate_FirstName()">
Last Name:<input type="text" name=lname onChange="Validate_LastName()"></p>
<P>E-mail Address:<input type="text" name=email onChange="Validate_Email()"></p>
<P>Comments::<Textarea name=comment
Rows="4"Cols="30"onfocus=document.form1.comment.select()>Enter Your
Comment</Textarea>
</p>
<p align="center"><input type="button" value=submit onClick="return
processForm()"></p>
</form>
</body>
</html>
```

Output:-



```
//Program for checking the browser type:
<html>
<head>
<title>Checking the browser type</title>
</head>
<body>
<script language="JavaScript">

if(navigator.appName=="Microsoft Internet Explorer")
        {
                document.write("<B><CENTER>")
                d\ocument.write("You have Internet Explorer " + navigator.appVersion)
                document.write("</B></CENTER>")
        }

if(navigator.appName=="Netscape")
        {
                document.write("<B><CENTER>")
                document.write("You have Netscape Navigator " + navigator.appVersion)
                document.write("</B></CENTER>")
        }
</script>
<CENTER>
        <H1>Checking Your Browser TYpe</h1>
</center>
</body>
</html>
```



(P.g.67) -2000

Tiling:- Background pictures are repeated under the contents of a page. This is called tiling bcoz each copy of the is like an identical tile on a kitchen floor. If you would like the background to be displayed once, Select the watermark option. The background won't scroll along with the rest of the page.

→ Timestamp: The time

Upasana

Q-1. Write HTML code using JvaScript.Inpt two number in text box and onClick of button it displays ***multiplication,addition,substraction and division*** of both numbers in third box.

```
<html>
<head>
<title>js</title>
<script type="text/Javascript">
<!--
    function mul()
    {
    var num1,num2,sum;
    num1=document.f1.n1.value ;
    num2=document.f1.n2.value ;
    sum=num1 * num2;
    document.f1.ss.value=sum;
    }
    function add()
    {
    var num1,num2,sum;
    num1=document.f1.n1.value*1 ;
    num2=document.f1.n2.value*1 ;
    sum=num1 + num2;
    document.f1.ss.value=sum;
    }
    function sub()
    {
    var num1,num2,sum;
    num1=document.f1.n1.value ;
    num2=document.f1.n2.value ;
    sum=num1 - num2;
    document.f1.ss.value=sum;
    }
    function div()
    {
    var num1,num2,sum;
    num1=document.f1.n1.value ;
    num2=document.f1.n2.value ;
    sum=num1 / num2;
    document.f1.ss.value=sum;
    }
//-->
</script>
<head>
<body>
<form name=f1 method=get>
Enter First Number:<input type=text name=n1><br>
```

```
Enter Second Number:<input type=text name=n2><br>
Result:<input type=text name=ss><br>
<input type=submit onClick=mul() value=mul>
<input type=submit onClick=add() value=Add>
<input type=submit onClick=sub() value=Sub>
<input type=submit onClick=div() value=divide>
<input type=reset >
</form>
</body>
</html>
```

(2)Write a JavaScript which displays current date and time.when page loads.

```
<html>
<head>
<title>dfd</title>
</head>
<body>
<script Language="Javascript">
var cdate=new Date()
document.write(cdate.getDate())
document.write("/")
document.write(cdate.getMonth())
document.write("/")
document.write(cdate.getFullYear())
document.write("<br>TIme</br>")
document.write(cdate.getHOurs()+ ": ")
document.write(cdate.getMinutes()+":"+cdate.getSeconds())
</script>
</body>
</html>
```

(3)Write a JavaScript.which takes a number through textboc display factorial of it in another text box on clicking on button.

```
<html>
<head>
<title>dfd</title>
</head>
<body>
<SCRIPT LANGUAGE=JavaScript>
var fact=1
for(var n=1;n<=10;n++)
{
        fact=fact * n;
        document.writeln(n + "!= " + fact +"<BR>")
}
```

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

(4)Write a Javascript,which takes 2 numer through 2 different textbox and display minimum number in third text box.

```
<html>
<head>
<title>dfd</title>
</head>
<body>
<SCRIPT LANGUAGE=JavaScript>
document.write("THe minimum of 1 and 2 is" + min(4.3,5))
function min(n1,n2)
{

      if(n1 < n2)
            {
            return(n1)
            }
      else
      {
            return(n2)
      }
}


</SCRIPT>
<H1>Using Fuction</h1>
</body>
</html>
```

(4)

```
<html>
<head>
<title>dfd</title>
<SCRIPT LANGUAGE=JavaScript>
var p=0
var msg="Welcome!Hi I am Upasana Mehta"
timerID=setInterval('showmsg()',200)

function showmsg()
{

    if(p>150)
    {
          p=0
```

```
        }
        var txt=" "
        for(var n=0;n<=p;n++)
        {
                txt=txt + " "
        }
        txt=txt + msg
        window.defaultStatus=txt
        p++
}
</SCRIPT>

</head>
<body>
<H1>Using Fuction</h1>
</body>
</html>
```

(5)

```
<html>
<head>
<title>did</title>
</head>
<body>
<SCRIPT LANGUAGE=JAVASCRIPT>
document.write(getCurrentTime())
function getCurrentTime()
{
        var time new Date
        var r="The Current Time is:" + time.getHours() +":" +time.getMinutes() +":"
+time.getSecond()
        return(r)
}


</SCRIPT>
<H1>Using Fuction</h1>
</body>
</html>
```

(6)Factorial:

```
<html>
<head>
<title>java scirpt</title>
<script type="text/javaScript">
<!--
```

Upasana

```
<html>
<head>
<title>Form Validation</title>
<script type="text/JavaScript">
<!--
    var whitespace=" \t\n\r";
    /*Define validations to run */

    validations = new Array();
    validations[0]=["document.form1.username","notblank"];
    validations[1]=["document.form1.useremail","validemail"];
    validations[2]=["document.form1.favnumber","isnumber"];

    function isEmpty(s)
    {
        var i;
        if((s==null) || (s.length==0))
        return true;
        //search string looking for characters that are not whitespace
        for(i=0;i<s.length;i++)
        {
            var c=s.charAt(i);
            if(whitespace.indexOf(c)==-1)
                return false;
        }
        //All characters are whitespace.
        return true;
    }
    function isEmail(field)
    {
        var pos;
        var s=field.value;
        if(isEmpty(s))
        {
            alert("Email may not be empty");
            field.focus();
            return false;
        }
        pos=s.indexOf('@',1);
        if((pos==-1) || (pos==(s.length-1)))
        {
            alert("Email not in valid form");
            field.focus();
            return false;
        }
        return true;
    }

    function isDigit(c)
    {
        return ((c>="0") && (c <="9"))
    }
```

```
function isInteger(field)
{
        var i,c;
        var s=field.value;
        if(isEmpty(s))
        {
                alert("Field cannot be empty");
                field.focus();
                return false;
        }
        for(i=0;i<s.length;i++)
        {
                //check if current character is number.
                c=s.charAt(i);
                if(!isDigit(c))
                {
                        alert("Filed must contain only digits");
                        field.focus();
                        return false;
                }
        }
        return true;
}

function validate()
{
        var i;
        var check;
        var field;
        for(i=0;i<validations.length;i++)
        {
                check=validations[i][1];
                field=eval(validations[i][0]);
                switch(check)
                {
                        case 'notblank':if(isEmpty(field.value))
                                        {
                                                alert("Field may not be empty");
                                                field.focus();
                                                return false;
                                        }
                                        break;
                        case 'validemail':if(!isEmail(field))
                                          return false;
                                          break;
                        case 'isnumber':if(!isInteger(field))
                                        return false;
                }
        }
                return true;
}
```

```
//-->
</script>
</head>
<body>
<form name="form1" method="get" action="val.html" onSubmit="return
validate();">
Username:<input type="text"name="username">
<br>
Email:<input type="text" name="useremail">
<br>
Favourite Number:<input type="text" name="favnumber">
<br>
<input type=submit>
</form>
</body>
</html>
```