Introduction to JavaScript

JavaScript is a scripting language developed by Netscape.

What is JavaScript?

- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript can be inserted into an HTML page
- JavaScript is an open scripting language that anyone can use without purchasing a license
- JavaScript is supported by all major browsers like Netscape and Internet Explorer

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

<script type="text/javascript">
document.write("<h1>Hello World!</h1>")
</script>

JavaScript Where To ...

Head section
Scripts that contain functions go in the head section of the document. Then we can be sure that the script is loaded before the function is called.

Body section
Execute a script that is placed in the body section.

External script
How to access an external script.

Sometimes you might want to run the same script on several pages, without writing the script on each and every page.

To simplify this you can write a script in an external file, and save it with a .js file extension,

```
<html>
<head>
</head>
<body>
<script src="xxx.js"></script>
</body>

</html>
```

<html>
<head>

```
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event")
}
</script>
</head>
<body onload="message()">
</body>
</html>
```

```
<body>
<script type="text/javascript">
document.write("This message is written when the page loads")
</script>
</body>
```

Variables

A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.

Rules for Variable names:

- Variable names are case sensitive
- They must begin with a letter or the underscore character

Declare a Variable

You can create a variable with the var statement:

```
var strname = some value
```

You can also create a variable without the var statement:

```
strname = some value
```

Assign a Value to a Variable

You assign a value to a variable like this:

```
var strname = "Hege"
```

Or like this:

```
strname = "Hege"
```

The variable name is on the left side of the expression and the value you want to assign to the variable is on the right. Now the variable "strname" has the value "Hege".

```
<body>

<script type="text/javascript">
var name = "Hege"
document.write(name)
document.write("<h1>"+name+"</h1>")
</script>

<p>This example declares a variable, assigns a value to it, and then displays the variable.</p>
```

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.

Operators are used to operate on values.

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	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 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 a 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!".

If and If...else Statement

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

```
var d=new Date()
var time=d.getHours()

if (time<10)
```

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

Switch Statement

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

```
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>
```

Looping

Very often when you write code, you want the same block of code to run a number of times.

while

The while statement will execute a block of code while a condition is true..

```
while (condition)
{
    code to be executed
}
```

```
<script type="text/javascript">
i = 0
while (i <= 5)
{
document.write("The number is " + i)
document.write("<br>")
i++
}
</script>
```

---

do...while

The do…while statement will execute a block of code once, and then it will repeat the loop while a condition is true

```
do
{
```

```
    code to be executed
}
while (condition)
```

```
<script type="text/javascript">
i = 0
do
{
document.write("The number is " + i)
document.write("<br>")
i++
}
while (i <= 5)
</script>
```

---

for

The for statement will execute a block of code a specified number of times

```
for (initialization; condition; increment)
{
    code to be executed
}
```

```
<script type="text/javascript">
for (i = 0; i <= 5; i++)
{
document.write("The number is " + i)
document.write("<br>")
}
</script>

<script type="text/javascript">
for (i = 1; i <= 6; i++)
{
document.write("<h" + i + ">This is header " + i)
document.write("</h" + i + ">")
}
</script>
```

Functions

A function contains some code that will be executed by an event or a call to that function. A function is a set of statements. You can reuse functions within the same script, or in other documents. You define functions at the beginning of a file (in the head section), and call them later in the document.

Some functions return a value to the calling expression

```
function result(a,b)
{
c=a+b
return c
}
```

The return Statement

Functions that will return a result must use the "return" statement. This statement specifies the value which will be returned to where the function was called from. Say you have a function that returns the sum of two numbers:

```
function total(a,b)
{
result=a+b
return result
}
```

When you call this function you must send two arguments with it:

```
sum=total(2,3)
```

The returned value from the function (5) will be stored in the variable called sum.

Symbols

Open symbols, like ({ [" ', must have a matching closing symbol, like ' "] }).

White Space

JavaScript ignores extra spaces. You can add white space to your script to make it more readable. These two lines mean exactly the same:

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

Break up a Code Line

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

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

Note: You can not break up a code line like this:

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

The example above will cause an error.

Insert Special Characters

You can insert special characters (like " ' ; &) with the backslash:

```
document.write ("You \& I sing \"Happy Birthday\".")
```

The example above will produce this output:

```
You & I sing "Happy Birthday".
```

Comments

You can add a comment to your JavaScript code starting the comment with two slashes "//":

```
sum=a + b  //calculating the sum
```

You can also add a comment to the JavaScript code, starting the comment with "/*" and ending it with "*/"

```
sum=a + b  /*calculating the sum*/
```

Using "/*" and "*/" is the only way to create a multi-line comment:

```
/* This is a comment
block. It contains
several lines*/
```

String Object

The fontcolor() method

var txt="W3Schools is great!!"
document.write(" " + txt.fontcolor('red') + " ")
document.write(" " + txt.fontcolor('blue') + " ")

JavaScript Window

<script type="text/javascript">
alert("Hello World!")
</script>

<script type="text/javascript">
var name = confirm("Press a button")
if (name == true)
{
document.write("You pressed OK")
}
else
{
document.write("You pressed Cancel")
}
</script>

```
<script type="text/javascript">
var name = prompt("Please enter your name","")
if (name != null && name != "")
{
document.write("Hello " + name)
}
</script>

<html>
<head>
```

Open a new window when clicking on a button

How you can display a new window.

```
<script language=javascript>
function openwindow()
{
window.open("../default.htm")
}
</script>

</head>
<body>
<form>
<input type=button value="Open Window" onclick="openwindow()">
</form>
</body>
</html>
```

Open a new window and control its appearance

How you can display a new window, but also decide the new window's appearance.

```
<script type="text/javascript">
function openwindow()
{
window.open("../default.htm","my_new_window","toolbar=yes,location=yes,directori
es=no,status=no,menubar=yes,scrollbars=yes,resizable=no,copyhistory=yes,width=40
0,height=400")
}
</script>

</head>
<body>
```

```
<form>
<input type="button" value="Open Window" onclick="openwindow()">
</form>

<html>
<head>
```

Multiple windows

How to pop up more than one window with just one click.

```
<script language=javascript>
function openwindow()
{
window.open("../../www.microsoft.com/default.htm")
window.open("../default.htm")
}
</script>

</head>
<body>
<form>
<input type=button value="Open Windows" onclick="openwindow()">
</form>
</body>
</html>

<html>
<head>
```

Location

How to send the client to a new location (URL/page).

```
<script type="text/javascript">
function locate()
{
location="../default.htm"
}
</script>

</head>
<body>

<form>
```

```
<input type="button" onclick="locate()" value="New location">
</form>

</body>
</html>
```

Status bar

How to write some text in the windows status bar.

```
<html>
<head>
<script type="text/javascript">
function load()
{
window.status = "put your message here"
}
</script>
</head>
<body onload="load()">
<p>Look in the statusbar</p>
</body>
</html>
```

Print page

How to print the page.

```
<script type="text/javascript">
function printpage()
{
window.print()
}
</script>

</head>
<body>

<form>
<input type="button" value="Print this page" onclick="printpage()">
</form>

</body>
</html>
```

Open and Close Windows

```
<html>
<head>
<script language=javascript>
function openwindow()
{
mywindow=window.open("../default.htm","My_new_window","width=300,height=3
00")
}
function closewindow()
{
mywindow.close()
}
</script>

</head>
<body>
<form>
<input type=button value="Open Window" onclick="openwindow()">
<input type=button value="Close Window" onclick="closewindow()">
</form>
</body>
</html>
```