

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language - a scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript is usually embedded directly in HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license
- JavaScript is supported by all major browsers, like Netscape and Internet Explorer

Are Java and JavaScript the same?

NO!

Java and JavaScript are two completely different languages!

Java (developed by Sun Microsystems) is a powerful and very complex programming language - in the same category as C and C++.

What can a JavaScript Do?

- **JavaScript gives HTML designers a programming tool** - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- **JavaScript can put dynamic text into an HTML page** - A JavaScript statement like this: `document.write("<h1>" + name + "</h1>")` can write a variable text into an HTML page
- **JavaScript can react to events** - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- **JavaScript can read and write HTML elements** - A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data** - A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing

How to Put a JavaScript Into an HTML Page

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

The code above will produce this output on an HTML page:

```
Hello World!
```

Example Explained

To insert a script in an HTML page, we use the `<script>` tag. Use the `type` attribute to define the scripting language

```
<script type="text/javascript">
```

Then the JavaScript starts: The JavaScript command for writing some output to a page is **document.write**

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

Then the `<script>` tag has to be closed

```
</script>
```

Ending Statements With a Semicolon?

With traditional programming languages, like C++ and Java, each code statement has to end with a semicolon.

Many programmers continue this habit when writing JavaScript, but in general, semicolons are **optional!** However, semicolons are required if you want to put more than one statement on a single line.

How to Handle Older Browsers

Browsers that do not support scripts will display the script as page content. To prevent them from doing this, we may use the HTML comment tag:

```
<script type="text/javascript">
<!--
    some statements
//-->
</script>
```

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

Note: You cannot put `//` in front of the first comment line (like `//<!--`), because older browsers will display it. Strange? Yes! But that's the way it is.

The HTML `<script>` tag is used to insert a JavaScript into an HTML page.

Examples

[Write text](#)

How to write text on a page.

```
<html>

<body>

<script type="text/javascript">

document.write("Hello World!")

</script>

</body>

</html>
```

Write text with formatting

How to format the text on your page with HTML tags.

```
<html>
<body>

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

</body>
</html>
```

Where to Put the JavaScript

Scripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

Scripts in the head section: Scripts to be executed when they are called, or when an event is triggered, go in the head section. When you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

```
<html>
<head>
<script type="text/javascript">
    some statements
</script>
</head>
```

Scripts in the body section: Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page.

```
<html>
```

```
<head>
</head>
<body>
<script type="text/javascript">
    some statements
</script>
</body>
```

Scripts in both the body and the head section: You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/javascript">
    some statements
</script>
</head>
<body>
<script type="text/javascript">
    some statements
</script>
</body>
```

How to Run an External JavaScript

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, like this:

```
document.write("This script is external")
```

Save the external file as xxx.js.

Note: The external script cannot contain the <script> tag

Now you can call this script, using the "src" attribute, from any of your pages:

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

Remember to place the script exactly where you normally would write the script.

Scripts in the body section will be executed WHILE the page loads.

Scripts in the head section will be executed when CALLED.

Examples

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.

```
<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 section

Execute a script that is placed in the body section.

```
<html>

<head>

</head>

<body>

<script type="text/javascript">

document.write("This message is written when the page loads")
```

```
</script>
```

```
</body>
```

```
</html>
```

External script

How to access an external script.

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<script src="xxx.js">
```

```
</script>
```

```
<p>
```

The actual script is in an external script file called "xxx.js".

```
</p>
```

```
</body>
```

```
</html>
```

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".

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.

JavaScript Operators

[← Previous](#) | [Next →](#)

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 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!".

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. It is now time to take a lesson about the alert-box:

This is JavaScript's method to alert the user.

```
alert("This is a message")
```

How to Define a Function

To create a function you define its name, any values ("arguments"), and some statements:

```
function myfunction(argument1, argument2, etc)  
{  
  some statements  
}
```

A function with no arguments must include the parentheses:

```
function myfunction()  
{  
  some statements  
}
```

Arguments are variables used in the function. The variable values are values passed on by the function call.

By placing functions in the head section of the document, you make sure that all the code in the function has been loaded before the function is called.

Some functions return a value to the calling expression

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

How to Call a Function

A function is not executed before it is called.

You can call a function containing arguments:

```
myfunction(argument1, argument2, etc)
```

or without arguments:

```
myfunction()
```

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.

A function is a reusable code-block that will be executed by an event, or when the function is called.

Examples

Function

How to call a function.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function myfunction()
```

```
{
```

```
alert("HELLO")
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
<input type="button"
```

```
onclick="myfunction()"
```

```
value="Call function">
```

```
</form>
```

```
<p>By pressing the button, a function will be called. The function will alert a message.</p>
```

```
</body>
```

```
</html>
```

Function with arguments

How to pass a variable to a function, and use the variable value in the function.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function myfunction(txt)
```

```
{
```

```
  alert(txt)
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>

<input type="button"
onclick="myfunction('Hello')"
value="Call function">

</form>
```

<p>By pressing the button, a function with an argument will be called. The function will alert this argument.</p>

```
</body>

</html>
```

Function with arguments 2

How to pass variables to a function, and use these variable values in the function.

```
<html>

<head>

<script type="text/javascript">

function myfunction(txt)

{

alert(txt)

}

</script>

</head>

<body>

<form>
```

```
<input type="button"
onclick="myfunction('Good Morning!')"
value="In the Morning">
```

```
<input type="button"
onclick="myfunction('Good Evening!')"
value="In the Evening">
</form>
```

```
<p>
```

When you click on one of the buttons, a function will be called. The function will alert the argument that is passed to it.

```
</p>
```

```
</body>
```

```
</html>
```

Function that returns a value

How to let the function return a value.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function myFunction()
```

```
{
```

```
return ("Hello, have a nice day!")
```

```
}  
</script>  
  
</head>  
<body>  
  
<script type="text/javascript">  
document.write(myFunction())  
</script>  
  
<p>The script in the body section calls a function.</p>  
  
<p>The function returns a text.</p>  
  
</body>  
</html>
```

A function with arguments, that returns a value

How to let the function find the sum of 2 arguments and return the result.

```
<html>  
<head>  
  
<script type="text/javascript">  
function total(numberA,numberB)  
{  
return numberA + numberB  
}  
</script>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write(total(2,3))
```

```
</script>
```

```
<p>The script in the body section calls a function with two arguments, 2 and 3.</p>
```

```
<p>The function returns the sum of these two arguments.</p>
```

```
</body>
```

```
</html>
```

Conditional Statements

Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In JavaScript we have three conditional statements:

- **if statement** - use this statement if you want to execute a set of code when a condition is true
- **if...else statement** - use this statement if you want to select one of two sets of lines to execute
- **switch statement** - use this statement if you want to select one of many sets of lines to execute

If and If...else Statement

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

Syntax

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

Example

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

if (time<10)
{
document.write("<b>Good morning</b>")
}
</script>
```

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

If you want to execute some code if a condition is true and another code if a condition is false, use the if...else statement.

Syntax

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

Example

```
<script type="text/javascript">
//If the time on your browser 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>
```

Switch Statement

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

Syntax

```
switch (expression)
```

```

{
case label1:
  code to be executed if expression = label1
  break
case label2:
  code to be executed if expression = label2
  break
default:
  code to be executed
  if expression is different
  from both label1 and label2
}

```

This is how it works: First we have a single expression (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>

```

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.

Conditional statements in JavaScript are used to perform different actions based on different conditions.

Examples

If statement

How to write an If statement. Use this statement if you want to execute a set of code if a specified condition is true.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var d = new Date()
```

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

```
if (time < 10)
```

```
{
```

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

```
}
```

```
</script>
```

```
<p>
```

```
This example demonstrates the If statement.
```

```
</p>
```

```
<p>
```

```
If the time on your browser is less than 10,
```

```
you will get a "Good morning" greeting.
```

```
</p>
```

```
</body>
```

```
</html>
```

If...else statement

How to write an If...Else statement. Use this statement if you want to execute one set of code if the condition is true and another set of code if the condition is false.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var d = new Date()
```

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

```
if (time < 10)
```

```
{
```

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

```
}
```

```
else
```

```
{
```

```
document.write("<b>Good day</b>")
```

```
}
```

```
</script>
```

```
<p>
```

This example demonstrates the If...Else statement.

```
</p>
```

```
<p>
```

If the time on your browser is less than 10,

you will get a "Good morning" greeting.

Otherwise you will get a "Good day" greeting.

```
</p>
```

```
</body>
```

```
</html>
```

Random link

This example demonstrates a link, when you click on the link it will take you to W3Schools.com OR to RefsnesData.no. There is a 50% chance for each of them.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var r=Math.random()
```

```
if (r>0.5)
```

```
{
```

```
document.write("<a href='http://www.w3schools.com'>Learn Web Development!</a>")
```

```
}
```

```
else
```

```
{
```

```
document.write("<a href='http://www.refsnedata.no'>Visit Refsnes Data!</a>")
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Switch statement

How to write a switch statement. Use this statement if you want to select one of many blocks of code to execute.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
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 really looking forward to this weekend!")
```

```
}
```

```
</script>
```

```
<p>This example demonstrates the switch statement.</p>
```

<p>You will receive a different greeting based on what day it is.</p>

<p>Note that Sunday=0, Monday=1, Tuesday=2, etc.</p>

</body>

</html>

Looping

Very often when you write code, you want the same block of code to run a number of times. You can use looping statements in your code to do this.

In JavaScript we have the following looping statements:

- **while** - loops through a block of code while a condition is true
- **do...while** - loops through a block of code once, and then repeats the loop while a condition is true
- **for** - run statements a specified number of times

while

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

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

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)
```

for

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

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

Looping statements in JavaScript are used to execute the same block of code a specified number of times.

Examples

For loop

How to write a For loop. Use a For loop to run the same block of code a specified number of times

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
for (i = 0; i <= 5; i++)
```

```
{
```

```
document.write("The number is " + i)
```

```
document.write("<br>")
```

```
}
```

```
</script>
```

```
<p>Explanation: </p>
```

```
<p>The for loop sets <b>i</b> equal to 0.</p>
```

```
<p>As long as <b>i</b> is less than , or equal to, 5, the loop will continue to run.</p>
```

```
<p><b>i</b> will increase by 1 each time the loop runs.</p>
```

```
</body>
```

```
</html>
```

Looping through HTML headers

How to use the For loop to write the HTML headers.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
for (i = 1; i <= 6; i++)
```

```
{
```

```
document.write("<h" + i + ">This is header " + i)
```

```
document.write("</h" + i + ">")
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

While loop

How to write a While loop. Use a While loop to run the same block of code while or until a condition is true

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
i = 0
```

```
while (i <= 5)
```

```
{
```

```
document.write("The number is " + i)
```

```
document.write("<br>")
```

```
i++
```

```
}
```

```
</script>
```

```
<p>Explanation: </p>
```

```
<p><b>i</b> equal to 0.</p>
```

```
<p>While <b>i</b> is less than , or equal to, 5, the loop will continue to run.</p>
```

```
<p><b>i</b> will increase by 1 each time the loop runs.</p>
```

```
</body>
```

```
</html>
```

Do while loop

How to write a Do While loop. Use a Do While loop to run the same block of code while or until a condition is true. This loop will always be executed once, even if the condition is false, because the statements are executed before the condition is tested

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
i = 0
```

```
do
```

```
{
```

```
document.write("The number is " + i)
```

```
document.write("<br>")
```

```
i++
```

```
}
```

```
while (i <= 5)
```

</script>

<p>Explanation:</p>

<p>i equal to 0.</p>

<p>The loop will run</p>

<p>i will increase by 1 each time the loop runs.</p>

<p>While i is less than , or equal to, 5, the loop will continue to run.</p>

</body>

</html>

JavaScript Guidelines

◀ Previous | Next ▶

Some things to know about JavaScript.

JavaScript is Case Sensitive

A function named "myfunction" is not the same as "myFunction". Therefore watch your capitalization when you create or call variables, objects and functions.

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 string** 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*/
```

Navigator Object

The Navigator object contains information about the client browser.

The Navigator object's collections, properties and methods are described below:

NN: Netscape, **IE:** Internet Explorer

Collections

Collection	Description	NN	IE
mimeTypes[]		3	4
plugins[]		3	4

Properties

Syntax: object.property_name

Property	Description	NN	IE
appName	The code name of the browser	2	3.02
appName	The name of the browser	2	3.02
appVersion	The version of the browser	2	3.02
cookieEnabled	A Boolean value that specifies whether cookies are enabled in the browser	6	4
platform	The operating system platform	4	4
userAgent	The value of the user-agent header sent by the client to the server	2	3.02

Methods

Syntax: object.method_name()

Method	Description	NN	IE
javaEnabled()	A Boolean value that specifies whether the browser has JavaScript enabled	3	4
refresh()		3	4
preference()		4	4
taintEnabled()	A Boolean value that specifies whether the browser has data tainting enabled	3	4

The Navigator object contains information about the client browser.

Examples

[Browser detection](#)

How to detect the client's browser

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
document.write("You are browsing this site with: " + navigator.appName)
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

[More details](#)

How to detect more details about the client

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write("<p>Browser: ")
```

```
document.write(navigator.appName + "</p>")
```

```
document.write("<p>Browser version: ")
```

```
document.write(navigator.appVersion + "</p>")
```

```
document.write("<p>Code: ")
```

```
document.write(navigator.appCodeName + "</p>")
```

```
document.write("<p>Platform: ")
```

```
document.write(navigator.platform + "</p>")
```

```
document.write("<p>Cookies enabled: ")
```

```
document.write(navigator.cookieEnabled + "</p>")
```

```
document.write("<p>Browser's user agent header: ")
```

```
document.write(navigator.userAgent + "</p>")
```

```
</script>
```

```
</body>
```

```
</html>
```

Redirect

How to redirect the user to different pages, depending on browser

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function redirectme()
```

```
{
```

```
bname=navigator.appName
```

```
if (bname.indexOf("Netscape")!=-1)
```

```
{
```

```
    window.location="tryjs_netscape.htm"
```

```
    return
```

```
}
```

```
if (bname.indexOf("Microsoft")!=-1)
```

```
{
```

```
    window.location="tryjs_microsoft.htm"
```

```
    return
```

```
}
```

```
window.location="tryjs_other.htm"
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onload="redirectme()">
```

```
</body>
```

```
</html>
```

Message

How to write a different message to the user, depending on the browser

```
<html>
<head>
<script type="text/javascript">
function browserversion()
{
txt="Your Browser is unknown"
browser=navigator.appVersion
if (browser.indexOf("2.")>-1)
{
txt="Your Browser is from the stone-age"
}
if (browser.indexOf("3.")>-1)
{
txt="You should update your Browser."
}
if (browser.indexOf("4.")>-1)
{
txt="Your Browser is good enough"
}
document.getElementById("message").innerHTML=txt
}
</script>
</head>

<body onload="browserversion()">
<span id="message"></span>
</form>

</body>
</html>
```

Screen Object

The Screen object is automatically created by the JavaScript runtime engine and it contains information about the client's display screen. Its properties are described below:

NN: Netscape, **IE:** Internet Explorer

Properties

Syntax: screen.property_name

Property	Description	NN	IE
----------	-------------	----	----

availHeight	The height of the display screen - excluding the Taskbar	4	4
availWidth	The width of the display screen - excluding the Taskbar	4	4
colorDepth	The bit depth of the color palette in use. If no color palette is in use it reflects the pixelDepth property	4	4
height	The height of the display screen	4	4
pixelDepth	The color resolution (in bits per pixel) of the display screen	4	
width	The width of the display screen	4	4

The Screen object contains information about the client's display screen.

Examples

Monitor

How to detect details about the client's monitor.

```
<html>
<body>
<script type="text/javascript">
document.write("SCREEN RESOLUTION: ")
document.write(screen.width + " * ")
document.write(screen.height + "<br>")
document.write("AVAILABLE VIEW AREA: ")
document.write(screen.availWidth + " * ")
document.write(screen.availHeight + "<br>")
document.write("COLOR DEPTH: ")
document.write(screen.colorDepth + "<br>")
</script>
</body>
</html>
```

Window Object

The Window object corresponds to the browser window. A Window object is created automatically with every instance of a <body> or <frameset> tag.

The Window object's properties, methods, and events are described below:

NN: Netscape, **IE:** Internet Explorer

Properties

Syntax: window.property_name

Property	Description	NN	IE
closed	Returns a Boolean value that specifies whether a window has been closed	3	4
defaultStatus	Defines the default text in the Windows' statusbar (will be displayed when the page loads)	2	3
document	The document object contained in the window	2	3

frames	Returns all the named frames in the window	2	3
history	Returns the URLs visited from the window	2	3
length	Returns the number of child frames in the window	6	4
location	Returns the URL of the window	2	3
name	Returns or sets the name of the window	2	3
navigator		6	4
opener	When using the window.open() method, you can use the opener property from the destination window to return details of the source window. Example: window.opener.close() will close the source window	3	3
parent	A reference to the window that contains the calling child frame	2	3
screen		6	4
self		2	3
status		2	3
top		2	3
window		2	3

Methods

Syntax: window.method_name()

Method	Description	NN	IE
alert("msg")	Displays an alert box with a message and an OK button	2	3
blur()	Removes focus from the current window	3	4
clearInterval(ID)	Cancels a timeout set with setInterval()	4	4
clearTimeout(ID)	Cancels a timeout set with setTimeout()	2	3
close()	Closes the current window	2	3
confirm("msg")	Displays a dialog box with a message, a Cancel, and an OK button	2	3
focus()	Sets focus on the current window	3	4
moveBy(x,y)	Moves the window a specified number of pixels in relation to its current co-ordinates	4	4
moveTo(x,y)	Moves the window's left and top edge to the specified x and y co-ordinates	4	4
open(URL,"name","specs")	Opens a new browser window		
print()	Prints the contents of the current window	4	5
prompt("msg","reply")	Displays a dialog box prompting the user for some input	2	3
scrollBy(x,y)	Scrolls the content by the specified number of pixels. Note: The visible property of the window's scrollbar must be set to true if using this method	4	4
scrollTo(x,y)	Scrolls the contents to the specified co-ordinates	4	4
setInterval(func,millsec)	Calls a function / evaluate an expression at the specified interval (in milliseconds)	4	4
setTimeout(func,millsec)	Calls a function / evaluate an expression after the specified number of milliseconds		
stop()	Cancels the current download. Same as the stop button in the browser	4	

Events

Syntax: window.event_name="someJavaScriptCode"

Event	Description	NN	IE
onBlur	Executes some code when a Blur event occurs	3	4
onError	Executes some code when an Error event occurs	3	4
onFocus	Executes some code when a Focus event occurs	3	4
onLoad	Executes some code when an Load event occurs	2	3
onResize	Executes some code when a Resize event occurs	4	4
onUnload	Executes some code when an Unload event occurs	2	3

The Window object corresponds to the browser window.

Examples

Alert box

How to display an alert box.

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

Confirm box

How to display a confirm box.

```
<html>
<body>

<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>
```

```
</body>
```

```
</html>
```

Prompt box

How to display a prompt box.

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var name = prompt("Please enter your name", "");
```

```
if (name != null && name != "")
```

```
{
```

```
document.write("Hello " + name)
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Open a new window when clicking on a button

How you can display a new window.

```
<html>

<head>

<script language=javascript>

function openwindow()

{

window.open("http://www.w3schools.com")

}

</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.

```
<html>

<head>

<script type="text/javascript">

function openwindow()

{
```

```
window.open("http://www.w3schools.com","my_new_window","toolbar=yes, location=yes,
directories=no, status=no, menubar=yes, scrollbars=yes, resizable=no, copyhistory=yes,
width=400, height=400")
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
<input type="button"
```

```
value="Open Window"
```

```
onclick="openwindow()">
```

```
</form>
```

```
</body>
```

```
</html>
```

Multiple windows

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

```
<html>
```

```
<head>
```

```
<script language=javascript>
```

```
function openwindow()
```

```
{
```

```
window.open("http://www.microsoft.com/")
```

```
window.open("http://www.w3schools.com/")
```

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

Location

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

```
<html>  
  
<head>  
  
<script type="text/javascript">  
  
function locate()  
  
{  
  
location="http://www.w3schools.com/"  
  
}  
  
</script>  
  
</head>  
  
<body>  
  
<form>  
  
<input type="button" onclick="locate()" value="New location">
```

```
</form>
```

```
</body>
```

```
</html>
```

Refresh

How to refresh a document.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function refresh()
```

```
{
```

```
location.reload()
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
<input type="button" value="Refresh" onclick="refresh()">
```

```
</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.

```
<html>

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

</head>
<body>

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

</body>
</html>
```

Document Object

The Document object is used to access all elements in a page.

The Document object's collections, properties, methods, and events are described below:

NN: Netscape, **IE:** Internet Explorer

Collections

Collection	Description	NN	IE
anchors[]	A reference to all Anchor objects in the document		
applets[]	A reference to all Applet objects in the document		
attributes[]			
childNodes[]			
embeds[]			
forms[]			
images[]	A reference to all Image objects in the document		
links[]			
plugins[]			
styleSheets[]			

Properties

Syntax: document.property_name

Property	Description	NN	IE
alinkColor	Sets or returns the color of the active links in the document	3	4
bgColor	Sets or returns the background-color of the document	3	4
cookie	Returns all cookies associated with the document		
domain	Returns the document server's Domain name	3	4
fgColor	Sets or returns the text-color of the document	3	4
lastModified	Returns the date and time the document was last modified	3	4
linkColor	Sets or returns the color of the links in the document	3	4
location			
referrer	Returns the URL of the document that loaded the current document	3	4
title	Returns the title of the document (text inside the HTML title element)	3	4
URL	Returns the URL of the current document	3	4
vlinkColor	Sets or returns the color of the visited links in the document	3	4

Methods

Syntax: document.method_name()

Method	Description	NN	IE
clear()			
close()			
focus()			
open("mimetype",replace)			
write("str")			
writeln("str")			

Events

Syntax: `document.event_name="someJavaScriptCode"`

Event	Description	NN	IE
onBlur	Executes some code when a Blur event occurs	2	3
onClick	Executes some code when a Click event occurs	2	3
onDbIcClick	Executes some code when a Doubleclick event occurs	2	3
onFocus	Executes some code when a Focus event occurs	2	3
onKeyDown	Executes some code when a Keydown event occurs	2	3
onKeyPress	Executes some code when a Keypress event occurs	2	3
onKeyUp	Executes some code when a Keyup event occurs	2	3
onMouseDown	Executes some code when a Mousedown event occurs	2	3
onMouseMove	Executes some code when a Mousemove event occurs	2	3
onMouseOut	Executes some code when a Mouseout event occurs	2	3
onMouseOver	Executes some code when a Mouseover event occurs	2	3
onMouseUp	Executes some code when a Mouseup event occurs	2	3
onResize	Executes some code when a Resize event occurs	2	3

String object

The String object is used to work with text.

The String object's properties and methods are described below:

NN: Netscape, **IE:** Internet Explorer

Properties

Syntax: `object.property_name`

Property	Description	NN	IE
constructor		4	4
length	Returns the number of characters in a string	2	3

Methods

Syntax: `object.method_name()`

Method	Description	NN	IE
<code>anchor("anchername")</code>	Returns a string as an anchor	2	3
<code>big()</code>	Returns a string in big text	2	3
<code>blink()</code>	Returns a string blinking	2	
<code>bold()</code>	Returns a string in bold	2	3
<code>charAt(index)</code>	Returns the character at a specified position	2	3
<code>charCodeAt(i)</code>	Returns the Unicode of the character at a specified position	4	4
<code>concat()</code>	Returns two concatenated strings	4	4
<code>fixed()</code>	Returns a string as teletype	2	3
<code>fontcolor()</code>	Returns a string in a specified color	2	3

fontSize()	Returns a string in a specified size	2	3
fromCharCode()	Returns the character value of a Unicode	4	4
indexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs	2	3
italics()	Returns a string in italic	2	3
lastIndexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs. Note: This method starts from the right and moves left!	2	3
link()	Returns a string as a hyperlink	2	3
match()	Similar to indexOf and lastIndexOf, but this method returns the specified string, or "null", instead of a numeric value	4	4
replace()	Replaces some specified characters with some new specified characters	4	4
search()	Returns an integer if the string contains some specified characters, if not it returns -1	4	4
slice()	Returns a string containing a specified character index	4	4
small()	Returns a string as small text	2	3
split()	Splits a string into an array of strings	4	4
strike()	Returns a string strikethrough	2	3
sub()	Returns a string as subscript	2	3
substr()	Returns the specified characters. 14,7 returns 7 characters, from the 14th character (starts at 0)	4	4
substring()	Returns the specified characters. 7,14 returns all characters from the 7th up to but not including the 14th (starts at 0)	2	3
sup()	Returns a string as superscript	2	3
toLowerCase()	Converts a string to lower case	2	3
toUpperCase()	Converts a string to upper case	2	3

The String object is used to work with text.

Examples

[The length property](#)

This example returns the number of characters in a string.

```
<html>

<body>

<script type="text/javascript">

var str="W3Schools is great!"

document.write("<p>" + str + "</p>")

document.write(str.length)

</script>
```

```
</body>
```

```
</html>
```

The fontcolor() method

This example returns a string in a specified color.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var txt="W3Schools is great!!"
```

```
document.write("<p>" + txt.fontcolor() + "</p>")
```

```
document.write("<p>" + txt.fontcolor('red') + "</p>")
```

```
document.write("<p>" + txt.fontcolor('blue') + "</p>")
```

```
document.write("<p>" + txt.fontcolor('green') + "</p>")
```

```
</script>
```

```
</body>
```

```
</html>
```

The indexOf() method

This example tests if a string contains a specified text and returns the position of where the text was found.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var str="W3Schools is great!"  
  
var pos=str.indexOf("School")  
  
if (pos>=0)  
{  
  
document.write("School found at position: ")  
  
document.write(pos + "<br />")  
  
}  
  
else  
  
{  
  
document.write("School not found!")  
  
}  
  
</script>
```

<p>This example tests if a string contains a specified word. If the word is found it returns the position of the first character of the word in the original string. Note: The first position in the string is 0!</p>

```
</body>
```

```
</html>
```

[The match\(\) method](#)

This example tests if a string contains a specified text and returns the text if it was found.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var str = "W3Schools is great!"
```

```
document.write(str.match("great"))
```

```
</script>
```

<p>This example tests if a string contains a specified word. If the word is found it returns the word.</p>

</body>

</html>

The substr() and substring() methods

This example returns a specified part of a string.

<html>

<body>

```
<script type="text/javascript">
```

```
var str="W3Schools is great!"
```

```
document.write(str.substr(2,6))
```

```
document.write("<br /><br />")
```

```
document.write(str.substring(2,6))
```

```
</script>
```

<p>

The substr() method returns a specified part of a string. If you specify (2,6) the returned string will be from the second character (start at 0) and 6 long.

</p>

<p>

The substring() method also returns a specified part of a string. If you specify (2,6) it returns all characters from the second character (start at 0) and up to, but not including, the sixth character.

</p>

</body>

```
</html>
```

The toLowerCase() and toUpperCase() methods

This example converts a string to lowercase and uppercase.

```
<html>
```

```
<body>
```

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

```
</body>
```

```
</html>
```

Array Object

The Array object is used to store a set of values in a single variable name. Each value is an element of the array and has an associated index number.

You create an instance of the Array object with the "new" keyword. The following example creates two arrays, both of three elements:

```
var family_names=new Array(3)  
var family_names=new Array("Tove","Jani","Stale")
```

You can refer to a particular element in the array by using the name of the array and the index number. The index number starts at 0.

If you create an array with a single numeric parameter, you can assign data to each of the elements in the array like this:

```
family_names[0]="Tove"  
family_names[1]="Jani"  
family_names[2]="Stale"
```

And the data can be retrieved by using the index of the particular array element you want. Like this:

```
mother=family_names[0]  
father=family_names[1]
```

The Array object's properties and methods are described below:

NN: Netscape, **IE:** Internet Explorer

Properties

Syntax: object.property_name

Property	Description	NN	IE
constructor	Contains the function that created an object's prototype	3	4
index		4	5.5
input		4	5.5
length	Represents the length of the array	3	4
prototype	Allows addition of properties to an array	3	4

Methods

Syntax: object.method_name()

Method	Description	NN	IE
concat()	Returns an array concatenated of two or more arrays	4	4
join()	Joins all the elements of an array into a string separated by a specified string separator (comma is default)	3	4
pop()	Removes and returns the last element of an array	4	5.5
push()	Adds one or more elements to the end of an array and returns the new length	4	5.5
reverse()	Reverses the order of the elements in an array	3	4
shift()	Removes and returns the first element of an array	4	5.5
slice()	Creates a new array from a selected section of an array	4	4
splice()	Used to add and/or remove elements of an array	4	5.5
sort()	Sorts the elements of an array	3	4
toSource()	Returns the source code of the array	4.06	4
toString()	Returns a string representing the specified array and its elements	3	4
unshift()	Adds one or more elements to the beginning of an array and returns the new length	4	5.5
valueOf()	Returns a primitive value for a specified array	4	3

The Array object is used to store a set of values in a single variable name.

Examples

[Create an array with a numeric parameter and assign data to it](#)

Create an array and assign data to it.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var famname = new Array(6)

famname[0] = "Jan Egil"

famname[1] = "Tove"

famname[2] = "Hege"

famname[3] = "Stale"

famname[4] = "Kai Jim"

famname[5] = "Borge"

for (i=0; i<6; i++)

{

document.write(famname[i] + "<br>")

}

</script>

</body>

</html>
```

Create a literal array

Create a literal array. Note that the length method is used to find out how many elements the array contains.

```
<html>

<body>

<script type="text/javascript">

var famname = new Array("Jan Egil", "Tove", "Hege", "Stale", "Kai Jim", "Borge")

for (i=0; i<famname.length; i++)

{

document.write(famname[i] + "<br>")
```

```
}  
</script>
```

```
</body>
```

```
</html>
```

Array - some properties and methods

Some properties and methods to use with the Array object.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var famname = new Array(3)
```

```
famname[0] = "Jani"
```

```
famname[1] = "Tove"
```

```
famname[2] = "Hege"
```

```
document.write(famname.length + "<br>")
```

```
document.write(famname.join(".") + "<br>")
```

```
document.write(famname.reverse() + "<br>")
```

```
document.write(famname.sort() + "<br>")
```

```
document.write(famname.push("Ola", "Jon") + "<br>")
```

```
document.write(famname.pop() + "<br>")
```

```
document.write(famname.shift() + "<br>")
```

```
</script>
```

```
</body>
```

```
</html>
```

Array - concat() and slice()

Some properties and methods to use with the Array object.

```
<html>
<body>

<script type="text/javascript">
var famname = new Array(3)
famname[0] = "Jani"
famname[1] = "Tove"
famname[2] = "Hege"

var famname2 = new Array(3)
famname2[0] = "John"
famname2[1] = "Andy"
famname2[2] = "Wendy"

document.write(famname.slice(2) + "<br>")
document.write(famname.concat(famname2) + "<br>")

</script>

</body>
</html>
```

Date Object

The Date object is used to work with dates and times.

You create an instance of the Date object with the "new" keyword.

To store the current date in a variable called "my_date":

```
var my_date=new Date()
```

After creating an instance of the Date object, you can access all the methods of the object from the "my_date" variable. If, for example, you want to return the date (from 1-31) of a Date object, you should write the following:

```
my_date.getDate()
```

You can also write a date inside the parentheses of the Date() object, like this:

```
new Date("Month dd, yyyy hh:mm:ss")
new Date("Month dd, yyyy")
new Date(yy,mm,dd,hh,mm,ss)
new Date(yy,mm,dd)
new Date(milliseconds)
```

Here is how you can create a Date object for each of the ways above:

```
var my_date=new Date("October 12, 1988 13:14:00")
var my_date=new Date("October 12, 1988")
var my_date=new Date(88,09,12,13,14,00)
var my_date=new Date(88,09,12)
var my_date=new Date(500)
```

The Date object's methods are described below:

NN: Netscape, **IE:** Internet Explorer

Methods

Syntax: object.method_name()

Method	Description	NN	IE
Date()	Returns a Date object	2	3
getDate()	Returns the date of a Date object (from 1-31)	2	3
getDay()	Returns the day of a Date object (from 0-6. 0=Sunday, 1=Monday, etc.)	2	3
getMonth()	Returns the month of a Date object (from 0-11. 0=January, 1=February, etc.)	2	3
getFullYear()	Returns the year of a Date object (four digits)	4	4
getYear()	Returns the year of a Date object (from 0-99). Use getFullYear instead !!	2	3
getHours()	Returns the hour of a Date object (from 0-23)	2	3
getMinutes()	Returns the minute of a Date object (from 0-59)	2	3
getSeconds()	Returns the second of a Date object (from 0-59)	2	3
getMilliseconds()	Returns the millisecond of a Date object (from 0-999)	4	4
getTime()	Returns the number of milliseconds since midnight 1/1-1970	2	3
getTimezoneOffset()	Returns the time difference between the user's computer and GMT	2	3
getUTCDate()	Returns the date of a Date object in universal (UTC) time	4	4
getUTCDay()	Returns the day of a Date object in universal time	4	4
getUTCMonth()	Returns the month of a Date object in universal time	4	4
getUTCFullYear()	Returns the four-digit year of a Date object in universal time	4	4
getUTCHours()	Returns the hour of a Date object in universal time	4	4
getUTCMinutes()	Returns the minutes of a Date object in universal time	4	4
getUTCSeconds()	Returns the seconds of a Date object in universal time	4	4
getUTCMilliseconds()	Returns the milliseconds of a Date object in universal time	4	4
parse()	Returns a string date value that holds the number of milliseconds since January 01 1970 00:00:00	2	3
setDate()	Sets the date of the month in the Date object (from 1-31)	2	3
setFullYear()	Sets the year in the Date object (four digits)	4	4
setHours()	Sets the hour in the Date object (from 0-23)	2	3
setMilliseconds()	Sets the millisecond in the Date object (from 0-999)	4	4
setMinutes()	Set the minute in the Date object (from 0-59)	2	3
setMonth()	Sets the month in the Date object (from 0-11. 0=January,	2	3

	1=February)		
setSeconds()	Sets the second in the Date object (from 0-59)	2	3
setTime()	Sets the milliseconds after 1/1-1970	2	3
setYear()	Sets the year in the Date object (00-99)	2	3
setUTCDate()	Sets the date in the Date object, in universal time (from 1-31)	4	4
setUTCDay()	Sets the day in the Date object, in universal time (from 0-6. Sunday=0, Monday=1, etc.)	4	4
setUTCMonth()	Sets the month in the Date object, in universal time (from 0-11. 0=January, 1=February)	4	4
setUTCFullYear()	Sets the year in the Date object, in universal time (four digits)	4	4
setUTCHour()	Sets the hour in the Date object, in universal time (from 0-23)	4	4
setUTCMinutes()	Sets the minutes in the Date object, in universal time (from 0-59)	4	4
setUTCSeconds()	Sets the seconds in the Date object, in universal time (from 0-59)	4	4
setUTCMilliseconds()	Sets the milliseconds in the Date object, in universal time (from 0-999)	4	4
toGMTString()	Converts the Date object to a string, set to GMT time zone	2	3
toLocaleString()	Converts the Date object to a string, set to the current time zone	2	3
toString()	Converts the Date object to a string	2	4

The Date object is used to work with dates and times.

Examples

Date

Returns today's date including date, month, and year. Note that the getMonth method returns 0 in January, 1 in February etc. So add 1 to the getMonth method to display the correct date.

```
<html>

<body>

<script type="text/javascript">

var d = new Date()

document.write(d.getDate())

document.write(".")

document.write(d.getMonth() + 1)

document.write(".")

document.write(d.getFullYear())
```

```
</script>
```

```
</body>
```

```
</html>
```

Time

Returns the current local time including hour, minutes, and seconds. To return the GMT time use `getUTCHours`, `getUTCMinutes` etc.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var d = new Date()
```

```
document.write(d.getHours())
```

```
document.write(" ")
```

```
document.write(d.getMinutes())
```

```
document.write(" ")
```

```
document.write(d.getSeconds())
```

```
</script>
```

```
</body>
```

```
</html>
```

Set date

You can also set the date or time into the date object, with the `setDate`, `setHour` etc. Note that in this example, only the `FullYear` is set.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var d = new Date()
```

```
d.setFullYear("1990")
```

```
document.write(d)
```

```
</script>
```

```
</body>
```

```
</html>
```

UTC time

The `getUTCDate` method returns the Universal Coordinated Time which is the time set by the World Time Standard.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var d = new Date()
```

```
document.write(d.getUTCHours())
```

```
document.write(" ")
```

```
document.write(d.getUTCMinutes())
```

```
document.write(" ")
```

```
document.write(d.getUTCSeconds())
```

```
</script>
```

```
</body>
```

```
</html>
```

Display weekday

A simple script that allows you to write the name of the current day instead of the number. Note that the array object is used to store the names, and that Sunday=0, Monday=1 etc.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```

var d=new Date()

var weekday=new
Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")

document.write("Today is " + weekday[d.getDay()])

</script>

</body>

</html>

```

Display full date

How to write a complete date with the name of the day and the name of the month.

```

<html>

<body>

<script type="text/javascript">

var d=new Date()

var weekday=new
Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")

var monthname=new
Array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")

document.write(weekday[d.getDay()] + " ")

document.write(d.getDate() + ". ")

document.write(monthname[d.getMonth()] + " ")

document.write(d.getFullYear())

</script>

</body>

</html>

```

Display time

How to display the time on your pages. Note that this script is similar to the Time example above, only this script writes the time in an input field. And it continues writing the time one time per second.

```

<html>
<head>
<script type="text/javascript">
var timer = null

```

```

function stop()
{
clearTimeout(timer)
}

function start()
{
var time = new Date()
var hours = time.getHours()
var minutes = time.getMinutes()
minutes=((minutes < 10) ? "0" : "") + minutes
var seconds = time.getSeconds()
seconds=((seconds < 10) ? "0" : "") + seconds
var clock = hours + ":" + minutes + ":" + seconds
document.forms[0].display.value = clock
timer = setTimeout("start()",1000)
}
</script>
</head>
<body onload="start()" onunload="stop()">
<form>
<input type="text" name="display" size="20">
</form>
</body>
</html>

```

Math Object

The built-in Math object includes mathematical constants and functions. You do not need to create the Math object before using it.

To store a random number between 0 and 1 in a variable called "r_number":

```
r_number=Math.random( )
```

To store the rounded number of 8.6 in a variable called "r_number":

```
r_number=Math.round( 8.6 )
```

The Math object's properties and methods are described below:

NN: Netscape, **IE:** Internet Explorer

Properties

Syntax: object.property_name

Property	Description	NN	IE
E	Returns the base of a natural logarithm	2	3
LN2	Returns the natural logarithm of 2	2	3
LN10	Returns the natural logarithm of 10	2	3
LOG2E	Returns the base-2 logarithm of E	2	3
LOG10E	Returns the base-10 logarithm of E	2	3
PI	Returns PI	2	3
SQRT1_2	Returns 1 divided by the square root of 2	2	3
SQRT2	Returns the square root of 2	2	3

Methods

Syntax: `object.method_name()`

Method	Description	NN	IE
<code>abs(x)</code>	Returns the absolute value of x	2	3
<code>acos(x)</code>	Returns the arccosine of x	2	3
<code>asin(x)</code>	Returns the arcsine of x	2	3
<code>atan(x)</code>	Returns the arctangent of x	2	3
<code>atan2(y,x)</code>	Returns the angle from the x axis to a point	2	3
<code>ceil(x)</code>	Returns the nearest integer greater than or equal to x	2	3
<code>cos(x)</code>	Returns the cosine of x	2	3
<code>exp(x)</code>	Returns the value of E raised to the power of x	2	3
<code>floor(x)</code>	Returns the nearest integer less than or equal to x	2	3
<code>log(x)</code>	Returns the natural log of x	2	3
<code>max(x,y)</code>	Returns the number with the highest value of x and y	2	3
<code>min(x,y)</code>	Returns the number with the lowest value of x and y	2	3
<code>pow(x,y)</code>	Returns the value of the number x raised to the power of y	2	3
<code>random()</code>	Returns a random number between 0 and 1	2	3
<code>round(x)</code>	Rounds x to the nearest integer	2	3
<code>sin(x)</code>	Returns the sine of x	2	3
<code>sqrt(x)</code>	Returns the square root of x	2	3
<code>tan(x)</code>	Returns the tangent of x	2	3

The built-in Math object includes mathematical constants and functions.

Examples

Round

How to round a specified number to the nearest whole number

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write(Math.round(7.25))
```

```
</script >
```

```
</body>
```

```
</html>
```

Random number

The random method returns a random number between 0 and 1

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write(Math.random())
```

```
</script >
```

```
</body>
```

```
</html>
```

Random number from 0 to 10

How to find a random number from 0 to 10 using the random() and round() methods.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
no=Math.random()*10
```

```
document.write(Math.round(no))
```

```
</script >
```

```
</body>
```

```
</html>
```

Max number

How to test which of two numbers, has the higher value.

```
<html>

<body>

<script type="text/javascript">

document.write(Math.max(2,4))

</script>

</body>

</html>
```

Min number

How to test which of two numbers, has the lower value.

```
<html>
<body>

<script type="text/javascript">
document.write(Math.min(2,4))
</script>

</body>
</html>
```

JavaScript Frames



Examples

Break out of frames

If you are trapped in a frame or two, this is how you break out.

```
<html>

<head>

<script type="text/javascript">
```

```
function breakout()
{
if (window.top != window.self)
{
window.top.location="tryjs_breakout.htm"
}
}
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

To break out of the frame:

```
<input type="button" onclick="breakout()" value="Click me">
```

```
</form>
```

```
</body>
```

```
</html>
```

Update 2 frames

How to update the 2 frames with one click.

```
<html>
```

```
<frameset rows="10%,*" frameborder="1">
```

```
    <frame name="upperframe" src="demo_js_frame_a.htm">
```

```
    <frame name="lowerframe" src="tryjs_twoframes_sourcepage.htm">
```

```
</frameset>
```

```
</html>
```

[Update 2 other frames](#)

How to update 2 frames from a 3rd. frame.

```
<html>
```

```
<frameset cols="80%,*" frameborder="1">
```

```
<frame name="leftframe" src="tryjs_threeframes_sourcepage.htm">
```

```
<frameset rows="30%,*" frameborder="1">
```

```
<frame name="upperframe" src="demo_js_frame_a.htm">
```

```
<frame name="lowerframe" src="demo_js_frame_b.htm">
```

```
</frameset>
```

```
</frameset>
```

```
</html>
```

JavaScript Form Object



Examples

[E-mail validation](#)

How you can validate an input-field that contains an e-mail address.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function validate()
```

```
{
```

```
x=document.myForm
```

```
at=x.myEmail.value.indexOf("@")

if (at == -1)
    {
        alert("Not a valid e-mail")
        return false
    }
}

</script>

</head>

<body>

<form name="myForm"
action="tryjs_submitpage.htm"
onsubmit="return validate()">

Enter your E-mail address:

<input type="text" name="myEmail">
<input type="submit" value="Send input">
</form>

<p>This example only tests if the email
address contains an "@" character.</p>

<p>Any "real life" code will have to test for
punctuations, spaces and other things as well.</p>

</body>
```

```
</html>
```

Value validation

How you can validate an input-field with min and max values.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function validate()
```

```
{
```

```
x=document.myForm
```

```
txt=x.myInput.value
```

```
if (txt>=1 && txt<=5)
```

```
{
```

```
return true
```

```
}
```

```
else
```

```
{
```

```
alert("Must be between 1 and 5")
```

```
return false
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form name="myForm" action="tryjs_submitpage.htm" onsubmit="return validate()">
```

Enter a value from 1 to 5:

```
<input type="text" name="myInput">
```

```
<input type="submit" value="Send input">
```

```
</form>
```

```
</body>
```

```
</html>
```

Length validation

How you can validate number of letters in an input-field.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function validate()
```

```
{
```

```
x=document.myForm
```

```
input=x.myInput.value
```

```
if (input.length>5)
```

```
{
```

```
alert("Do not insert more than 5 characters")
```

```
return false
```

```
}
```

```
else
{
return true
}
}
</script>
</head>

<body>

<form name="myForm" action="tryjs_submitpage.htm" onsubmit="return validate()">
```

In this input box you are not allowed to insert more than 5 characters:

```
<input type="text" name="myInput">

<input type="submit" value="Send input">

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

Form validation

A form containing all of the validation above.

```
<html>

<head>

<script type="text/javascript">

function validate()
```

```
{  
  
x=document.myForm  
  
at=x.myEmail.value.indexOf("@")  
  
code=x.myCode.value  
  
firstname=x.myName.value  
  
submitOK="True"  
  
if (at== -1)  
  
  {  
  
    alert("Not a valid e-mail")  
  
    submitOK="False"  
  
  }  
  
if (code<1 || code>5)  
  
  {  
  
    alert("Your code must be between 1 and 5")  
  
    submitOK="False"  
  
  }  
  
if (firstname.length>10)  
  
  {  
  
    alert("Your name must be less than 10 letters")  
  
    submitOK="False"  
  
  }  
  
if (submitOK=="False")  
  
  {  
  
    return false  
  
  }  
  
}  
  
</script>
```

```
</head>
```

```
<body>
```

```
<form name="myForm" action="tryjs_submitpage.htm" onsubmit="return validate()">
```

Enter your e-mail:

```
<input type="text" name="myEmail">
```

```
<br>
```

Enter your code, value from 1 to 5:

```
<input type="text" name="myCode">
```

```
<br>
```

Enter your first name, max 10 letters:

```
<input type="text" name="myName">
```

```
<br>
```

```
<input type="submit" value="Send input">
```

```
</form>
```

```
</body>
```

```
</html>
```

Focus

How to set focus on an input field.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function setfocus()
```

```
{
document.forms[0].field.focus()
}
</script>
</head>
<body>
<form>
<input type="text" name="field" size="30">
<input type="button" value="Get Focus" onclick="setfocus()">
</form>
</body>
</html>
```

Selected

How to make the content in an input field selected.

```
<html>
<head>
<script type="text/javascript">
function setfocus()
{
document.forms[0].field.select()
document.forms[0].field.focus()
}
</script>
</head>
<body>
<form>
<input type="text" name="field" size="30" value="input text">
```

```
<input type="button" value="Selected" onclick="setfocus()">
</form>
</body>
</html>
```

Radio button

How the client can select options from radio buttons.

```
<html>
<head>
<script type="text/javascript">
function check(browser)
{
document.forms[0].answer.value=browser
}
</script>

</head>
<body>

<form>

Which browser is your favorite<br>

<input type="radio"
name="browser" onclick="check(this.value)"
value="Explorer">Microsoft Internet Explorer<br>

<input type="radio"
name="browser" onclick="check(this.value)"
value="Netscape">Netscape Navigator<br>
```

```
<input type="text" name="answer">
```

```
</form>
```

```
</body>
```

```
</html>
```

Checkbox

How the client can select options from checkboxes.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function check()
```

```
{
```

```
coffee=document.forms[0].coffee
```

```
answer=document.forms[0].answer
```

```
txt=""
```

```
for (i = 0; i<coffee.length; ++ i)
```

```
{
```

```
if (coffee[i].checked)
```

```
{
```

```
txt=txt + coffee[i].value + " "
```

```
}
```

```
}
```

```
answer.value="You ordered a coffee with " + txt
```

```
}
```

```
</script>
```

```
</head>

<body>

<form>

How would you like your coffee?<br>

<input type="checkbox"

name="coffee" value="cream">With cream<br>

<input type="checkbox"

name="coffee" value="sugar">With sugar<br>

<input type="text" name="answer" size="30">

<input type="button" name="test" onclick="check()" value="Order">

</form>

</body>

</html>
```

Select from a dropdown list

How the client can select options from a drop down list.

```
<html>

<head>

<script type="text/javascript">

function put()

{

option=document.forms[0].dropdown.options[document.forms[0].dropdown.selectedIndex].text
```

```
txt=option

document.forms[0].favorite.value=txt
}
</script>

</head>

<body>

<form>

<p>

Select your favorite browser:

<select name="dropdown" onchange="put()">

<option>Internet Explorer

<option>Netscape Navigator

</select>

</p>

<p>

Your favorite browser is:

<input type="text"

name="favorite" value="Internet Explorer">

</p>

</form>

</body>

</html>
```

Select more than one option

How the client can select many options from a drop down list.

```
<html>

<head>

<script type="text/javascript">

function put()

{

option=document.forms[0].dropdown.options[document.forms[0].dropdown.selectedIndex].text

txt=document.forms[0].number.value

txt=txt + option

document.forms[0].number.value=txt

}

</script>

</head>

<body>

<form>

Select numbers: <br>

<select name="dropdown">

<option>1

<option>2

<option>3

<option>4

<option>5

<option>6

<option>7
```

```
<option>8
```

```
<option>9
```

```
<option>0
```

```
</select>
```

```
<input type="button" onclick="put()" value="-->"> <input type="text" name="number">
```

```
</form>
```

```
</body>
```

```
</html>
```