WEB HOSTING

Best Web Hosting

PHP MySQL Hosting

Best Hosting Coupons

UK Reseller Hosting

Cloud Hosting

Top Web Hosting

\$7.95/mo SEO Hosting

WEB BUILDING

Download XML Editor

FREE Website BUILDER

Free Website Templates

Free WordPress Themes

XML Editor « try me!

W3SCHOOLS EXAMS

Get Certified in: HTML, CSS, JavaScript,

XML, PHP, and ASP

W3SCHOOLS BOOKS

New Books:

HTML, CSS

JavaScript, and Ajax

STATISTICS

Browser Statistics Browser OS

Browser Display

SHARE THIS PAGE

Share with »

Next Chapter »

HOME HTML CSS XML JAVASCRIPT ASP PHP SQL MORE ...

REFERENCES | EXAMPLES | FORUM | ABOUT

JS Basic

JS HOME JS Introduction JS How To JS Where To JS Statements JS Comments JS Variables JS Operators JS Comparisons JS If...Else JS Switch

JS Popup Boxes

JS Functions JS For Loop JS While Loop

JS Break Loops JS For...In JS Events

JS Try...Catch JS Throw

JS Special Text JS Guidelines

JS Objects

JS Objects Intro JS Strina JS Date JS Array JS Boolean JS Math JS RegExp

JS Advanced

JS Browser JS Cookies

JS Validation

JS Timing JS Create Object JS Summary

JS Examples

JS Examples JS Objects Examples JS Browser Examples JS DOM Examples JS Quiz

JS Certificate

JS References

JavaScript Objects HTML DOM Objects

JavaScript Functions

« Previous

A function will be executed by an event or by a call to the function.

JavaScript Functions

To keep the browser from executing a script when the page loads, you can put your script into a function.

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

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

Functions can be defined both in the <head> and in the <body> section of a document. However, to assure that a function is read/loaded by the browser before it is called, it could be wise to put functions in the <head> section.

How to Define a Function

Syntax

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

The parameters 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.

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.

JavaScript Function Example

```
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>
 Try it yourself »
```

If the line: alert("Hello world!!") in the example above had not been put within a function, it would have been executed as soon as the page was loaded. Now, the script is not executed before a user hits the input button. The function displaymessage() will be executed if the input button is clicked.

You will learn more about JavaScript events in the JS Events chapter.

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.

The example below returns the product of two numbers (a and b):

```
Example
<html>
<head>
<script type="text/javascript">
function product(a,b)
return a*b;
</script>
</head>
<body>
<script type="text/javascript">
document.write(product(4,3));
</script>
</body>
</html>
 Try it yourself »
```

The Lifetime of JavaScript Variables

If you declare a variable, using "var", 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.



« Previous

More Examples

Function with a parameter

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

<u>Function that returns a value</u>

How to let a function return a value.

Next Chapter »

