A function is a set of JavaScript statements that performs a routine. Each function must have a name, that enables it to be called into action from other parts of the script. Functions can be called as many times as needed during the running of the script. For example, suppose you want to install a rollover image on an HTML page. You could write the function that performs that task as many times as you need it for the page. That could run into a lot of coding. However, it is better to write the function once and merely call it into action when you need it.

A function consists of the word *function* followed by the function name. The word *function* alerts the JavaScript that it is just that. After assigning it a name — any name that does not conflict with the JavaScript reserved keyword, you include a set of parentheses followed by an opening brace. The statements making up the function go on the following lines, then the function ends with a closing brace. Indentations are for easy reading and are optional. Here is what it might look like:

```
function doSomething() {
     document.write("<p>God bless America!</p>")
     }
```

It is handy to write the function as a head script. That way is can be called into action in the body whenever you want it. Although you may write the function, unless a script calls it nothing will come of it. You call it into action with an event handler somewhere else on the page. Here's an example:

```
<body onLoad="doSomething()">
```

The `href` attribute uses the pound sign (#) because the link does not go anywhere. The pound sign allows the JavaScript to work in the anchor tag.

Sometimes you may want to take some information and give it to the function for processing. This is referred to as *passing* the information. To make this work, the function is changed to something like the following:

```
function doSomething(msg) {
     alert(msg)
     }
```

The function would be located as a head script. To call the function, we can have an input object in the body of the document:

```
<a href="#" onclick="doSomething()">Click for message</a>
```

The `msg` variable in the function is a *parameter* or *argument*. When the function is called, the data `God bless America!` is passed into the `msg` parameter. Function parameters can be almost any data you want to use, including strings, numbers, or even other JavaScript objects.

Copyright © 2010 by Paul Roberts • All rights reserved

The foregoing function would allow you to reuse it over and over. Here are some examples that refer to the same head function:

```
<input type="button" value="Abraham Lincoln" onclick="doSomething('Four score and seven years ago...')">

<input type=”button” value="John Kennedy" onclick="doSomething('Ask not what your country can go for you...')">

<input type=”button” value="Richard Nixon" onclick="doSomething('They won't have Nixon to kick around anymore.')">

<input type=”button” value="Barack Obama" onclick="doSomething('Faith is not just something you have, it's something you do.')">
```

Each time the function is called, the input object passes the message to the `msg` parameter in the function. The function then returns the message designated by the input object.

Functions can be as simple or as complicated as you want. As the function becomes more sophisticated, you must be sure that you have matching braces for each object in the function.