

# HTML INTERFACE

## *The HTML - Javascript interface*

As with any programming language, or natural language for that matter, the best and perhaps the only way to learn it, is to dive in and practice. Work through examples, or create you own simple examples to understand what the proram statements mean. Before undertaking these examples, it is assumed you have at least a working knowledge of HTML and preferably **Cascading Style sheets** (CSS). I.e. you can produce a web page using nothing more complicated than a text editor (notepad or gedit for example). The reason being, this is probably where you will write and debug your Javascripts programs. Cascading style sheets, and the style command in particular, are what makes HTML accessible to Javascript.

## *Adding a script to the page*

To insert JavaScript into a web page, use the `<script>` tag. You should use the type attribute to specify the type of script being used, which in the case of JavaScript is *text/javascript*. It is also possible to the language attribute to say what JavaScript version you are using. In practice, this number means very little to browsers. They may claim to support a specific version, but will have vastly different capabilities.

Browsers will generally choose an arbitrary version number that they will claim to support, and will run any script that has either no language version number, or a version equal to or lower than the one they claim to support. Since the language version is so unreliable, you should generally omit this attribute, although my may see some scripts using it. Your script can then detect if the browser is capable of running the script lot more accurately than the version number can.

This is an example of a script tag, used to include a script in your page:

```
<script type="text/javascript">  
    //JavaScript goes here  
</script>
```

If your script is needed to access the page, or for the page to function correctly, you should include support for non-JavaScript browsers by using:

```
<noscript>  
  <p>This will not be displayed if  
    JavaScript is enabled</p>  
</noscript>
```

It may seem obvious, but the noscript tags need to be placed somewhere inside the body tags, and preferably at the top of the page where the user will see the message. **Don't** put the noscript tags in the header, because whether Javascript is enabled or not, the user won't see the message.

### ***Guidelines to adding JavaScript to a page***

You can add a script anywhere inside the head or body sections of your document. However, to keep your document well structured there are some basic guidelines:

- Most scripts can go inside the document head. This keeps them out of the way of the main document content.
- If your script needs to run at a certain stage during page layout (for example, if it uses document.write to create content), it should be put at the correct stage of the document, almost always somewhere inside

the document body. If the script is very small, then put it all where it will be used. If it is larger, then put it inside the head inside a function, and you can then call that function when you need it.

- If your script is used on more than one page, or if it is of any significant size, then put it in its own file, and load it in the document head.

## ***Using external script files***

You can also use a separate header file to contain the JavaScript (usually \*.js) making it easy to share across pages:

```
<script src="myExternalScript.js" type="text/javascript"></script>
```

Scripts in header files are executed as if they were in the main page. If the script references any external files, such as images, the addresses it uses are relative to the main page, not the script URI. The above example assumes the script is in the same folder as the web page.

## ***Hello world - A first Javascript program***

Its traditional that your first program is a *Hello world* program, to demonstrate that you can get the machine to talk to you. So in keeping with tradition, first build a simple HTML page to hold your Javascript and which will act as a template for further examples. This part of the process should be familiar to you.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>javascript test page</title>
<link href="./allStyles.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">
  /* this is where your javascript starts */
```

```
</script>
</head>
<body>
<h2>Subject</h2>
<h3>Introduction</h3>
<!-- This is where your HTML code starts -->
<h4>Program output</h4>
<div id="output">
</div>
</body>
</html>
```

Perhaps the first thing to note are the `<script>` tags. As written here, the assumption is that the code which follows is Javascript, the default language of the web.

### *Javascript and JScript*

Microsoft have their own version of Javascript (called JScript) and their own language version called VBScript. Neither are compatible with other browsers. E.g. Microsoft's JScript event model.

VBScript follows the conventions of Microsoft's Visual Basic. If you have a strong affinity for Visual basic and want to extend that into web design then you must use the language property of the script tag to specify the language being used. E.g.

```
<SCRIPT Language="VBScript">
```

**Note:** Using VBScript as your scripting language will tie you in to the Microsoft Explorer web browser, as currently this is the only browser that supports VBScript.

Now add the following Javascript code between the <SCRIPT> tags. You don't need to make the words *function* and *alert* bold. The bold style has been used to indicate these are reserved Javascript words.

```
function test()
{
    // Welcome greeting for new users
    alert("Hello World");
}
```

The curly braces { } signify the start { and end } of a code block. In this example they mark the start and end of the **function** called test. The function name is unimportant, but it ought at least to indicate what the function does.

To run the function example, create a button as shown, following the h2 header tag. You may add any additional text of your own, by way of explanation. When programming Javascript, or any other language for that matter, use plenty of comments to explain what is going on.

If you have renamed the function, make sure the button calls the right function

```
<form>
  <input type="button" value = "test" onclick = "test()" />
</form>
```



Javascript is case sensitive, this cannot be emphasised enough! So var my\_Variable is not the same as my\_variable. Check your spelling as well.

Source : [http://www.soslug.org/wiki/javascript\\_example\\_templates](http://www.soslug.org/wiki/javascript_example_templates)