

JavaScript

A script is a small piece of program that can add interactivity to the website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be written using JavaScript or VBScript.

JavaScript is an Event Driven Language

JavaScript is an event driven language which means your scripts react to events you set up. The code isn't running all the time; it simply waits until an event starts something up. These events can be attached to most any HTML tag (as an HTML attribute) or form element (as a method attached to the element). Below a short-list of common events:

Event	Description
onFocus	User just moved into this form element.
onBlur	A form field lost the focus (User moved to another field)
onAbort	An image failed to load.
onChange.	The contents of a field have changed.
onClick	User clicked on this item.
onDbClick.	User double-clicked on this item.
onKeyDown	A key was pressed.
onKeyPress	A key was pressed OR released.
onKeyUp	A key was released.
onMouseDown	A mouse button was pressed.
onMouseMove	The mouse moved.
onMouseOut	A mouse moved off of this element.
onMouseOver	The mouse moved over this element.
onSelect	Text has been selected.
onSubmit	A form's Submit button has been pressed.

onClick Event

The onClick event occurs when the user clicked on this item.

Example (Lucky number)

```
<!doctype html>
<html>
<head>
<title> Fun with Buttons</title>
<script type="text/javascript">
```

function cool()

```
{  
  var num = Math.floor((Math.random() * 100) + 1);  
  document.write("The lucky number for the day is " + num);  
}  
</script>  
</head>  
<body>  
<input type="button" value="Click for Lucky Number" onClick="cool();" />  
</body>  
</html>
```

onMouseOut and onMouseOver Event

The onMouseOut and onMouseOver events occur when a mouse moved off or over of element.

Example (change the color of the text)

```
<!doctype html> (Note that event be attached to HTML attribute)  
<html>  
<head>  
<title>Change Text Color</title>  
</head>  
<body>  
<h1 onmouseover="style.color='red'" onmouseout="style.color='pink'">Mouse over this text</h1>  
<p onmouseover="style.color='green'" onmouseout="style.color='blue'">Mouse over this text</p>  
</body> </html>
```

onfocus Event

The onfocus event occurs when an element gets focus. The onfocus event is most often used with <input>, <select>, and <a>.

Example

```
<!doctype html>  
<html>  
<head>  
<title>my page</title>  
</head>  
<body>  
Enter your name: <input type="text" id="f1" onFocus = "fFocus()">
```

```
<script>
function fFocus()
{

    var x = document.getElementById("f1");
    x.style.background = "yellow";
}
</script>
</body>
</html>
```

Onblur Event

The onblur event occurs when an object loses focus. The onblur event is most often used with form validation code (e.g. when the user leaves a form field).

Example

```
<!doctype html>
<html>
<head>
<title>My Site</title>
</head>
<body>
Enter your name: <input type="text" id="f2" onblur="fBlur()" >
<script>
function fBlur()
{
    var x = document.getElementById("f2");
    x.value = x.value.toUpperCase();
}
</script>
</body>
</html>
```

Add JavaScript to a Page

Javascript code can keep in a separate file and then include it wherever it's needed, or the code can define functionality inside HTML document itself.

- **Internal Script**

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using **<script>** tag, otherwise there is no restriction and you can put your source code anywhere in the document but inside **<script>** tag.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Internal Script</title>
<script type="text/javascript">
function Hello()
{
    alert("You are Winner");
}
</script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body> </html>
```

- **External Javascript**

If you are going to define a functionality which will be used in various HTML documents then it's better to keep that functionality in a separate Javascript file and then include that file in your HTML documents. A Javascript file will have extension as **.js** and it will be included in HTML files using **<script>** tag.

Example

Using separate Javascript file **script.js** which has following code:

```
function Hello()
{
    alert("You are Winner"); // or document.write("You are Winner ");
}
```

Then in the HTML document:

```
<!DOCTYPE html>
<html>
```

```
<head>
<title>External Script</title>
<script src="script.js" type="text/javascript" /> </script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```

Accessing nodes

The document object in the Document Object Model (DOM) identifies the page itself; The DOM is a collection of nodes (Element nodes, Attribute nodes and Text nodes). The document object comes with a number of standard and built-in properties and methods for accessing collections of elements. There are several methods for accessing nodes in the document:

- **By element name**

getElementsByTagName ()

Example **var paragraphs = document.getElementsByTagName("p");**

This method retrieves any element or elements you specify as an argument. For example, document.getElementsByTagName("p") returns every paragraph on the page.

- **By class attribute value**

getElementsByClassName ()

Example **var firstColumn = document.getElementsByClassName("column");**

This method allows you to access nodes in the document based on the value of a class attribute. The example statement assigns any element with a class value of “column” to the variable firstColumn so it can be accessed easily from within a script.

- **By id attribute value**

getElementById ()

Example

```

var photo = document.getElementById("lead-photo");
```

This method returns a single element based on that element’s ID (the value of its id attribute), which we provide to the method as an argument.

Example (access by id attribute value)

```
<!doctype html>
<html>
<head>
<title>My Site</title>
</head>
<body>
<h1 id="example" onClick="color()">Click on this text to change the background color</h1>
<br><br>
<p id="example2" onmouseover ="color2()">Click on this text to change the background color</p>
<script type='text/javascript'>
function color()
{
    var x = document.getElementById("example");
    x.style.backgroundColor='pink';
    x.style.textAlign='center';
}
function color2()
{
    var y = document.getElementById("example2");
    y.style.backgroundColor='red';
    y.style.textAlign='center';
}
</script>
</body>
</html>
```

JavaScript for Form Validation

Form validation used to occur at the server, after the client had entered all necessary data and then pressed the Submit button. If some of the data that had been entered by the client had been in the wrong form or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process and over burdening server.

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

- **Basic Validation** : First of all, the form must be checked to make sure data was entered into each form field that required it. This would need just loop through each field in the form and check for data.
- **Data Format Validation** : Secondly, the data that is entered must be checked for correct form and value. This would need to put more logic to test correctness of data.

Form Validation - Checking for Non-Empty

Example

```
<!doctype html>
<html>
<head>
<title>My Page</title>
</head>
<body>
<script type='text/javascript'>
function notEmpty(elem)
{
    if(elem.value.length == 0)
    {
        alert('Please Enter a Value');
        elem.focus();
        return false;
    }
    return true;
}
</script>
<form>
Required Field: <input type='text' id='req1' />
<input type='button' onclick="notEmpty(document.getElementById('req1'))" value='Check Field' />
</form>
</body> </html>
```

Form Validation - Restricting the Length**Example**

```
<!doctype html>
<html>
<head>
<title>my page</title>
</head>
<body>
<script type='text/javascript'>
function lengthRestriction(elem, min, max)
{
    var input = elem.value;
    if(input.length >= min && input.length <= max)
    {
        return true;
    }
    else
    {
        alert("Please enter between " +min+ " and " +max+ " characters");
        elem.focus();
        return false;
    }
}
</script>
<form>
Username(input 6-8 characters): <input type='text' id='f3'/>
<input type='button' onclick="lengthRestriction(document.getElementById('f3'), 6, 8)"
    value='Check Field' />
</form>
</body>
</html>
```

Form Validation - Selection Made**Example**

```
<!doctype html>
<html>
<head>
<title>my page</title>
```



```
</head>
<body>
<script type='text/javascript'>
function madeSelection(elem)
{
    if(elem.value == "Please Choose")
    {
        alert('Please Choose Something');
        elem.focus();
        return false;
    }
    else
    {
        return true;
    }
}
</script>
<form>
Selection: <select id='selection'>
<option>Please Choose</option>
<option>CA</option>
<option>WI</option>
<option>XX</option>
</select>
<input type='button'
    onclick="madeSelection(document.getElementById('selection'))" value='Check Field' />
</form>
</body>
</html>
```

Example

```
<!doctype html>
<html>
<head>
<title>my page</title>
</head>
<body>
<script type='text/javascript'>
function madeSelection(elem)
{
```

```
        if(elem.value == -1)
        {
            alert('Please Choose Something');
            elem.focus();
            return false;
        }
    else
    {
        return true;
    }
}
</script>
<form>
Country: <select id='selection'>
<option value="-1" selected>choose your country</option>
<option value="1">IRAQ</option>
<option value="2">UK</option>
<option value="3">USA</option>

</select>
<input type='button'
        onclick="madeSelection(document.getElementById('selection'))" value='Check Field' />
</form>
</body>
</html>
```

Form Validation - Checking for Numbers

Example

```
<!doctype html>
<html>
<head>
<title>my page</title>
</head>
<body>
<script type='text/javascript'>
function isNumeric(elem)
{
    var numericExpression = /^[0-9]+$/;
    if(elem.value.match(numericExpression))
    {
```

```
        return true;
    }
    else
    {
        alert('Numbers Only Please');
        elem.focus();
        return false;
    }
}
</script>
<form>
Numbers Only: <input type='text' id='numbers' />
<input type='button' onclick="isNumeric(document.getElementById('numbers'))" value='Check Field' />
</form>
</body>
</html>
```

Form Validation - Checking for All Letters

Example

```
<!doctype html>
<html>
<head>
<title>My Site</title>
</head>
<body>
<script type='text/javascript'>
function isAlphabet(elem)
{
    var alphaExp = /^[a-zA-Z]+$/;
    if(elem.value.match(alphaExp))
    {
        return true;
    }
    else
    {
        alert('Letters Only Please');
        elem.focus();
        return false;
    }
}
```

```
}  
</script>  
<form>  
Letters Only: <input type='text' id='letters'/>  
<input type='button' onclick="isAlphabet(document.getElementById('letters'))"  
    value='Check Field' />  
</form>  
</body>  
</html>
```

Form Validation - Checking for Numbers and Letters

Example

```
<!doctype html>  
<html>  
<head>  
<title>My Site</title>  
</head>  
<body>  
<script type='text/javascript'>  
function isAlphabet(elem)  
{  
    var alphaExp = /^[0-9a-zA-Z]+$/;  
    if(elem.value.match(alphaExp))  
    {  
        return true;  
    }  
    else  
    {  
        alert('Letters and number Only Please');  
        elem.focus();  
        return false;  
    }  
}  
</script>  
<form>  
Letters Only: <input type='text' id='letters'/>  
<input type='button' onclick="isAlphabet(document.getElementById('letters'))" value='Check Field' />  
</form>  
</body>  
</html>
```

Form Validation - Email Validation

Every email is made up for 5 parts:

1. A combination of letters, numbers, periods, hyphens, plus signs, and/or underscores
2. The at symbol @
3. A combination of letters, numbers, hyphens, and/or periods
4. A period
5. The top level domain (com, net, org, us, gov, ...)

Valid Examples:

- bobby.jo@filltank.net
- jack+jill@hill.com
- the-stand@steven.king.com

Invalid Examples:

- @deleted. (no characters before the @)
- [free!dom@bravehe.art](#) (invalid character !)
- [shoes@need_shining.com](#) (underscores are not allowed in the domain name)

Example

```
<!doctype html>
<html>
<head>
<title>my page</title>
</head>
<body>
<script type='text/javascript'>
function email(elem)
{
    var emailExp = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
    // also can use emailExp =    /^[\\w\\-\\.\\+]+@[a-zA-Z0-9\\.\\-]+\\.[a-zA-z0-9]{2,4}$/;
    if(elem.value.match(emailExp))
    {
        return true;
    }
    else
    {
        alert('Not a Valid Email');
```

```
elem.focus();

return false;
}
}
</script>
<form>
Email: <input type='text' id='emailer' />
<input type='button' onclick="email(document.getElementById('emailer'))" value='Check Field' />
</form>
</body>
</html>
```

The regular expression for email is `/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/`

To understand the regular expression we will divide it into smaller components:

`/^[a-zA-Z0-9._-]+`: Means that the email address must begin with alpha-numeric characters (both lowercase and uppercase characters are allowed). It may have periods,underscores and hyphens.

`@`: There must be a '@' symbol after initial characters.

`[a-zA-Z0-9.-]+`: After the '@' sign there must be some alpha-numeric characters. It can also contain period ('.') and and hyphens('-').

`\.`: After the second group of characters there must be a period ('.'). This is to separate domain and subdomain names.

`[a-zA-Z]{2,4}$`: Finally, the email address must end with two to four alphabets. Having a-z and A-Z means that both lowercase and uppercase letters are allowed.

`{2,4}` indicates the minimum and maximum number of characters. This will allow domain names with 2, 3 and 4 characters e.g.; us, tx, org, com, net, wxyz).