



# Input Validation with JavaScript

## 3

### Introduction

This exercise will show you how to use JavaScript to perform simple interactive input validation in a browser using the HTML Document Object Model.

### The Exercise

There are a number of tasks required in this exercise. Each task is detailed below.

### Setting Up

Make a new directory for this exercise. Call this directory *Valid*:

```
C:\> mkdir Valid
C:\> cd Valid
```

All the files that you subsequently create as you do this exercise should be contained in this directory.

### Creating the HTML/Script Source

Create the following in a file called *Input Validation.html*:

```
<HTML>
<HEAD>
  <TITLE>JavaScript Form Input Validation Example</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    function tester()
    {
      var ssnOK = testSSN(theForm.SSN.value);
      var nameOK = testName (theForm.name.value);
      theForm.submitter.disabled = ! (ssnOK && nameOK);
    }
    function testSSN(val)
    {
      var isNum = isNumberString (val);
      ssntxt.style.color = isNum ? "" : "red";
      return (isNum);
    }
    function testName(v)
    {
      var isValid = v.length > 0;
      nametxt.style.color = isValid ? "" : "red";
      return (isValid);
    }
    function isNumberString (s)
    {
      if (s.length == 0) return (false);
      for (i = 0; i < s.length; i ++)
      {
        var c = s.substring (i, i + 1);
        if ("0123456789".indexOf (c, 0) == -1)
          return (false);
      }
      return (true);
    }
    function doSubmit ()
    {
      results.innerText = theForm.name.value + " " + theForm.SSN.value;
    }
  </SCRIPT>
</HEAD>
<BODY>
  <H1>JavaScript Form Input Validation Example</H1>
  <P>
    The first field accepts any characters as input, but MUST contain data.
    The SSN field will only accept numbers. The 'Submit' button will be
    enabled ONLY if both fields contain valid data.
  </P>
  <P>
```

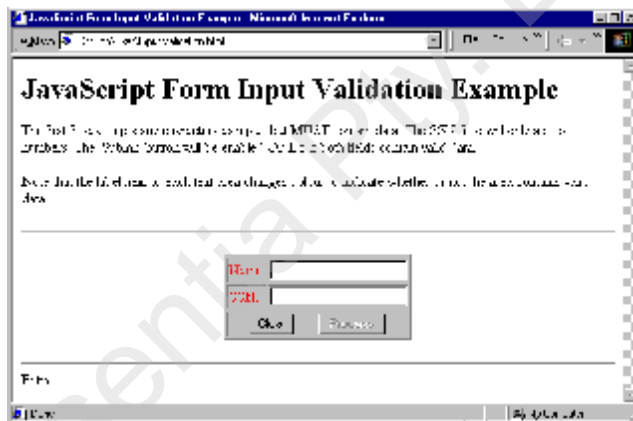
```

Note that the label next to each text area changes colour to indicate
whether or not the area contains valid data.
</P>
<HR>
<FORM NAME="theForm">
  <CENTER>
    <TABLE BORDER = "1" BGCOLOR="silver">
      <TR>
        <TD STYLE="COLOR:red" ID="nametxt">Name:</TD>
        <TD><INPUT NAME="name" onkeyup="tester()"></TD>
      </TR>
      <TR>
        <TD STYLE="COLOR:red" ID="ssntxt">SSN:</TD>
        <TD><INPUT TYPE="TEXT" NAME="SSN" SIZE="20"
          MAXLENGTH="40" onkeyup="tester()"></TD>
      </TR>
      <TR>
        <TD ALIGN="CENTER" COLSPAN="2">
          <INPUT TYPE="BUTTON" VALUE="Clear"
            onclick="document.theForm.reset();tester()">
          &nbsp;
          &nbsp;
          &nbsp;
          <INPUT TYPE="BUTTON" VALUE="Process" ID="submitter" DISABLED="TRUE"
            onclick="doSubmit()">
        </TD>
      </TR>
    </TABLE>
  </CENTER>
</FORM>
<HR>
Entry: <SPAN ID="results" STYLE="COLOR:blue"></SPAN>
</BODY>
</HTML>

```

### Executing the Resulting File

Open the Input Validation.html file in IE. You will see the page appear as in the following screenshot:



Play around with the dialog and be sure that you understand what is happening and how JavaScript is allowing you to provide a 'richer' user interface experience than is possible with 'pure' HTML.

### Closing Comment

A form such as this would normally send the resulting validated data to a web server for further use. To keep the requirements for this exercise to a minimum, I have omitted this capability.

The form can be easily adapted to send data to a server. The necessary changes are:

1. Modify the FORM tag to specify that data should be POSTed to the CGI script located at the URL <http://some.host.name/cgi-bin/input.pl>:



```
<FORM NAME="theForm" METHOD="POST"
      ACTION="http://some.host.name/cgi-bin/input.pl">
```

## 2. Modify the definition of the "Process" button to become:

```
<INPUT TYPE="SUBMIT" ID="submitter" DISABLED="TRUE" VALUE="Process">
```

## 3

With these simple changes in place, collected and validated data can be sent across the internet.

A simple CGI script suitable for processing the data transmitted by this script appears below:

```
#!/usr/bin/perl

require './cgi-lib.pl';

print &PrintHeader;

if (&ReadParse(*input))
{
    $name = $input{'name'};
    $SSN = $input{'SSN'};

    print <<EOF
    <HEAD>
    <TITLE>Echoing Entered Data...</TITLE>
    </HEAD>
    <BODY>
    <H1>Data sent to this form</H1>
    Name: <STRONG>$name</STRONG>
    <BR>
    SSN: <STRONG>$SSN</STRONG>
    </BODY>
    EOF
}
```