6. JavaScript Overview & JavaScript Form Validation

JavaScript Overview

It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages.

Advantages of JavaScript:

- Less server interaction
- Immediate feedback to the visitors
- Increased interactivity
- Richer interfaces

Examples:

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- Below example uses the method to "find" an HTML element (with id="demo") and changes the element content (innerHTML) to "Hello JavaScript":
  document.getElementById("demo").innerHTML = "Hello JavaScript";
- You can use an alert box to display data:
  ex: alert("test");
- For debugging purposes, you can use the console.log() method to display data in console. you can see output by opening consol window of browser.
  ex: console.log("test")
- JavaScript Variables:
  ex: var a = 6;
- JavaScript Operators: JavaScript uses arithmetic operators (+ - * /) to compute values:
  ex: (5 + 6) * 10
- JavaScript Functions: A JavaScript function is a block of code designed to perform a particular task.
  ex:

```
function myFunction(p1, p2) {
    return p1 * p2;              // The function returns the product of p1 and p2
}
```

- JavaScript Objects: Objects are variables too. But objects can contain many values.
  ex: var car = {type: “Fiat", model:"500", color: “white"};
- JavaScript Events:

  Common HTML Events

  Here is a list of some common HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

  example: <button onclick="displayDate()">The time is? </button>
- JavaScript Arrays: JavaScript arrays are used to store multiple values in a single variable.
  ex: var cars = ["Saab", "Volvo", "BMW"];

- There are also some JavaScript array methods available for doing basic array operations:
  ex: Popping and Pushing
  The pop() method removes the last element from an array:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.pop();              // Removes the last element ("Mango") from fruits
```

  Same way following operations for arrays are available: Shifting Elements, Changing Elements, Deleting Elements, Splicing an Array, Merging (Concatenating) Arrays, Slicing an Array, Automatic toString(), Finding Max and Min Values in an Array, Sorting etc.
- JavaScript Date Objects:
  ex: var d = new Date();
  output: Mon Oct 15 2018 17:53:04 GMT+0530 (India Standard Time)

JavaScript Form Validation

In most of scenarios form controls needs to be validated to ensure no false entries are submitted on server. there might be server-side validation but to improve user experience it is recommended to validate form on client side as well.

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server.

Example

```
<form action="/cgi-bin/test.cgi" name="myForm" onsubmit="return(validate());">
   <table cellspacing="2" cellpadding="2" border="1">

      <tr>
         <td align="right">Name</td>
         <td><input type="text" name="Name" /></td>
      </tr>

      <tr>
         <td align="right">EMail</td>
         <td><input type="text" name="EMail" /></td>
      </tr>

      <tr>
         <td align="right">Zip Code</td>
         <td><input type="text" name="Zip" /></td>
      </tr>

      <tr>
         <td align="right">Country</td>
         <td>
            <select name="Country">
               <option value="-1" selected>[choose yours]</option>
               <option value="1">USA</option>
               <option value="2">UK</option>
               <option value="3">INDIA</option>
            </select>
         </td>
      </tr>

      <tr>
         <td align="right"></td>
         <td><input type="submit" value="Submit" /></td>
      </tr>

   </table>
</form>
```

Here, we have added one form with basic html controls like textbox, dropdown etc. Now we want to ensure that when form is submitted no blank entries submitted also zip code length must be of 5 char long.

```
<script type="text/javascript">
   <!--
      // Form validation code will come here.
      function validate()
      {

         if( document.myForm.Name.value == "" )
         {
            alert( "Please provide your name!" );
            document.myForm.Name.focus() ;
            return false;
         }

         if( document.myForm.EMail.value == "" )
         {
            alert( "Please provide your Email!" );
            document.myForm.EMail.focus() ;
            return false;
         }

         if( document.myForm.Zip.value == "" ||
         isNaN( document.myForm.Zip.value ) ||
         document.myForm.Zip.value.length != 5 )
         {
            alert( "Please provide a zip in the format #####." );
            document.myForm.Zip.focus() ;
            return false;
         }

         if( document.myForm.Country.value == "-1" )
         {
            alert( "Please provide your country!" );
            return false;
         }
         return( true );
      }
   //-->
</script>
```

Here, function validate() has checks that ensures that no controls have blank entries and zip code is 5 char long only then form is submitted otherwise it returns false and shows error message