Lab 3: Javascript

Prepared: TrangNTT

Note: Using Firebug to debug Javascript.

11.1. Array

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<BODY>

<SCRIPT type="text/javascript">


// Create array and add three elements
var myArray = new Array();
myArray.push("Tim");
myArray.push("Janice");
myArray.push("Joe");

// print out the contents and newline
document.writeln( myArray.toString() );
document.writeln("<BR>");

// access array element by index
document.writeln( myArray[1] );
document.writeln("<BR>");

// print out the length of the array and newline
document.writeln( "Array Size: " + myArray.length );
document.writeln("<BR>");

// Pop the last element off the array and display it
var who = myArray.pop();
document.writeln( "Popped Element: " + who );
document.writeln("<BR>");

// print out the length of the array and newline
document.writeln( "Array Size: " + myArray.length );
document.writeln("<BR>");
document.writeln("<BR>");

// ***** //

// Populating array on instantiation
var otherArray = new Array("dog", 3, 8.4);

document.writeln( otherArray.toString() );
document.writeln("<BR>");
document.writeln("<BR>");

// setting and accessing associative arrays.
otherArray["key"] = "value";
document.writeln( otherArray["key"] );
document.writeln("<BR>");

</SCRIPT>

</BODY>
</HTML>
```

Result:

Tim,Janice,Joe
Janice
Array Size: 3
Popped Element: Joe
Array Size: 2

dog,3,8.4

value

11.2. Open new window

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<BODY>

<SCRIPT type="text/javascript">

if (window.confirm("Open Google?")) {
    newWindow = window.open('http://www.google.com','NewGoogleWindow','toolbar=no,alwaysRaised=yes');
    newWindow.focus();
}

</SCRIPT>

</BODY>
</HTML>
```

11.3. Count the Function calls

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>

    <SCRIPT type="text/javascript">

    num = 0;
    // Count the number of this function has been called. Print the result to a text field.
    function myFunc( ) {
        num++;
        document.myForm.outputField.value = "Called " + num + " times";
    }

    window.setInterval("myFunc()", 1000);

    </SCRIPT>

</HEAD>
<BODY>

<FORM name="myForm">
    <INPUT type="text" name="outputField" readonly>
</FORM>



</BODY>
</HTML>
```

11.4. Set controls' attributes

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>

    <SCRIPT type="text/javascript">

    function setAlign( align ) {
        var pTag = window.document.getElementById("paragraph");
        pTag.setAttribute("align", align);

    }

    </SCRIPT>

</HEAD>
<BODY>

<FORM name="alignForm">
    <SELECT id="alignSelect">
        <OPTION value="left">left</OPTION>
        <OPTION value="right">right</OPTION>
        <OPTION value="center">center</OPTION>
    </SELECT>
    <INPUT type="button" value="Go" onclick="setAlign( document.alignForm.alignSelect.value )">
</FORM>

<P id="paragraph">
    This is some sample text.
</P>

</BODY>
</HTML>
```

11.5. Add text fields

```
<html>
<head>
  <title>Example Message Box Page</title>
  <script type="text/javascript">
        var inputs = 1;

    function addField()
    {
        inputs++;

        var form = document.getElementById('sweet-form');
        var line_break = document.createElement('br');
        var textbox = document.createElement('input');

        textbox.setAttribute('type', 'text');
        textbox.setAttribute('name', 'text_'+inputs);
        textbox.setAttribute('value', 'I am box #'+inputs);

        form.appendChild(line_break);
        form.appendChild(textbox);
        return false;
    }//end addField

    function doLoad()
    {
      document.getElementById('sweet-link').addEventListener('click', addField, false);
    }//doLoad
    window.addEventListener('load', doLoad, false);
  </script>
</head>
<body>
    <form id="sweet-form">
        <input type="text" name="text_1" value="I am box #1"/>
    </form>
  <a id="sweet-link" href="#">Add Text Field</a>
</body>
</html>
```

11.6. Validate PostalCode and Money (Regular Expression)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>

    <SCRIPT type="text/javascript">

    function validatePostalCode( ) {
        var postalCodeCheck = new RegExp("([A-Z][0-9]){3,3}")
        if ( postalCodeCheck.test(document.myForm.postalCodeField.value) ) {
            alert("Valid Postal Code");
            return true;
        } else {
            alert("Invalid Postal Code: " + document.myForm.postalCodeField.value);
            return false;
        }
    }

    function validateMoney( ) {
        var moneyCheck = new RegExp("[0-9]*[\.][0-9][0-9]")
        if ( moneyCheck.test(document.myForm.moneyField.value) ) {
            alert("Valid Money");
            return true;
        } else {
            alert("Invalid Money: " + document.myForm.moneyField.value);
            return false;
        }
    }

    </SCRIPT>

</HEAD>
<BODY>

<FORM id="myForm">
    Postal Code: <INPUT type="text" name="postalCodeField" ><INPUT type="button"
                                    value="Check Postal Code" onClick="validatePostalCode()">
    <BR>
    Money: $<INPUT type="text" name="moneyField" ><INPUT type="button"
                                    value="Check Money" onClick="validateMoney()">
</FORM>



</BODY>
</HTML>
```

11.7. Exercise 1 - Event Handling

Write a simple HTML document called `colors.html` containing three colored links at the top of the page labeled RED, GREEN, and BLUE. Under the links there should be a small paragraph of text (you make this up) with a starting color of BLACK. Using `onMouseOver` and `onMouseOut` events of the links, when a user rolls over the each and any of the color labeled links, the color of the text in the paragraph below should change to the color corresponding to the link the user is hovering over. When the user rolls off the link, the color of the text should go back to it's starting color of BLACK.

11.8. Exercise 3 - DOM

Weight: 20/40

The purpose of this part of the assignment is to create a HTML document that enables a user to enter and save his/her favorite blog. The document consists of two separate frames. When the user fills out the form in *right* frame and submits it, the information submitted is saved in a cookie on the user's machine *and* the information is displayed in given fields in the *left* frame. If the user closes the browser window and re-opens the HTML document within twenty-four (24) hours, the information originally entered should still be displayed in the *left* frame because it will be stored in a cookie. Now suppose a user has a new favorite blog, therefore the user should be able to overwrite the existing information by simply entering new information into the form. Both the information in the *left* frame AND the cookie should be updated, so if the user re-opens the browser at a later time, only the most recent favorite blog is visible in the *left* frame.

You are provided with the following:

1. `index.html`: The `FRAMESET` document.
2. `left.html`: Where the user's favorite blog is displayed.
3. `right.html`: Where the user enters the favorite blog.

You need to:

1. Finish the `addBlog()` function in `right.html` to update the *left* frame and store the blog information in a cookie.
2. Add whatever code is needed to `left.html` so that the appropriate cookie is retrieved and the blog information displayed when the frame is opened or refreshed.

11.9. Exercise 4 – Validate inputs

Create a register form and validate inputs in the form (username… not null, email, telephone in the right format…)