INTERNET & WEB APPLICATION DEVELOPMENT SWE 444

Fall Semester 2008-2009 (081)

Module 3 (V): Client-Side Scripting (JavaScript)

Dr. El-Sayed El-Alfy
Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives/Outline

- Objectives
  - Understand the form object
  - Learn how to validate the form input using regular expressions
  - Learn how to define new objects and use them

- Outline
  - The Form Object
  - Validating Form Input
  - User Defined Objects
  - Script Debugging

The Form Object

- Forms are used to collect data from users and submit it for processing
  - Used to interact with a Web page and through which server and browser scripts respond to user needs
  - Web forms contain various types of **controls** like textbox, button, text area, drop-down list, etc.
- Form controls, as a group, often are enclosed inside **<form>** tags.
  - <form> tags can contain **action** and **method** attributes governing submission of form values for processing by server scripts.
  - <form> tags are *not* required when form controls are used for input to *browser* scripts.
- Example:

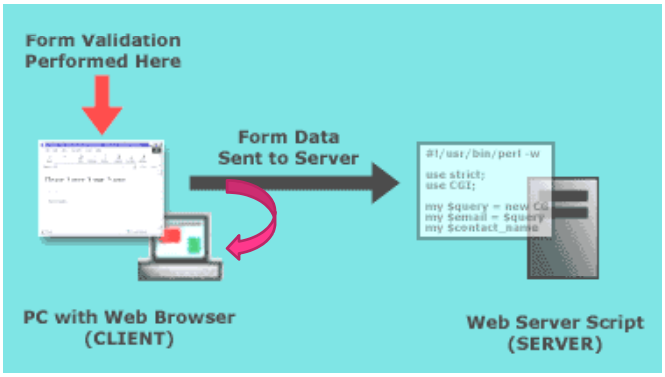
```
<input type="text" value="Change this text."
 onchange="document.getElementById("MSG").innerText=
      'You changed the text.'"/>
<span id="MSG"></span>
```


Validating Form Data

- JavaScript can be used to validate input data in XHTML forms before sending off the content to a server.
- Form data are typically checked by a JavaScript to see, e.g. :
  - has the user left required fields empty?
  - has the user entered a valid e-mail address?
  - has the user entered a valid date?
  - has the user entered text in a numeric field?

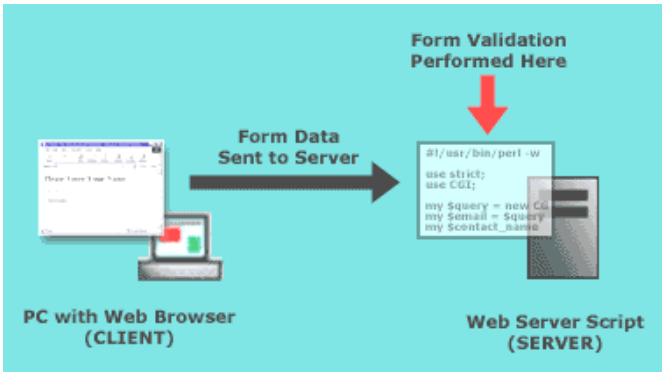
Validating Form Data (cont.)

- Server-side form validation



Validating Form Data (cont.)

- Server-side form validation



Example: Validating Empty Field

```
<script type="text/javascript">
<!--
function CheckNull(){
  document.getElementById("MSG").innerHTML = "";
  if (document.getElementById("MyField").value == "") {
    document.getElementById("MSG").innerHTML = "Missing data!";
    document.getElementById("MyField").focus();
  }
  else {
     document.getElementById("MyField").focus();
  }
}
//-->
</script>
</head>
<body>
<input type="Text" id="MyField"/>
<input type="button" value="Submit" onclick="CheckNull()"/>
<body onload="document.getElementById('MyField').focus()" />
<span id="MSG"></span>
```


Regular Expressions

- A notational convention used to match a word, a number or any other string of text within another
- Introduced in JavaScript 1.2 and mainly used in form validation
- A regular expression can be written statically or dynamically
  - Within slashes, such as `re = /ab+c/`
  - With a constructor, such as `re = new RegExp("ab+c")`
    - `Used when pattern to match is taken as user input, for example`
- Regular expressions are almost the same as in Perl or Java
  - only a few unusual features are missing
- Examples
  - var pattern = /[0-9]/
    - Matches an integer
  - var pattern = /[A-Za-z]/
    - Matches a string of letters
- Can specify more complex regular expressions to match phone numbers of the form `abc-def-ghij` where `a, b, …, j` are digits

Special Characters in Regular Expressions

Token	Description
^	Match at the start of the input string
$	Match at the end of the input string
*	Match 0 or more times
+	Match one or more times
?	Match 0 or 1 time
a|b	Match a or b
{n}	Match the string n times
\d	Match a digit
\D	Match a non-digit

Cont.

Token	Description
\w	Match any alphanumeric character or underscore
\W	Match anything except alphanumeric characters or underscores
\s	Match a whitespace character
\S	Match anything except for whitespace characters
[…]	Creates a set of characters, one of which must match if the operation is to be successful. If you need to specify a range of characters then separate the first and the last with a hypen: [0-9] or [P-X]
[^…]	Creates a set of characters which must not match. If any character in the set matches then the operation has failed. This fails if any lowercase letter d to q is matched: [^d-q]

- Examples: http://home.cogeco.ca/~ve3ll/jstutorf.htm

Example: Validating ID and Password

```
function is8CharacterString(elm){
    var pattern = /[A-Za-z0-9]/;
    if(pattern.test(elm.value))
         return true;
    else return false;
}
function isReady(iField, pField) {
    var iResult = iField.value.length != 6?false:is6DigitInt(iField);
    var pResult = pField.value.length != 8?false:is8CharacterString(pField);
    if (iResult == false) {
        alert("Please enter 6-digit integer.");
        iField.focus();
        return false;
    }
    if (pResult == false) {
        alert("Please enter 8-character string.");
        pField.focus();
        return false;
    }
    return true;
}
//-->
</script>
```


Cont.

```
<script language="JavaScript">
<!--
function is6DigitInt(elm) {
    if (elm.value == "") {
        return false;
    }
    for (var i = 0; i < elm.value.length; i++) {
        if (elm.value.charAt(i) < "0" || elm.value.charAt(i) > "9“){
            return false;
        }
    }
    return true;
}
function is6DigitInt2(elm){
    var pattern = /[^0-9]/;
    if(pattern.test(elm.value))
         return false;
    else return true;
}
// continued …
```

Example: Validating Phone and E-Mail

```
function isReady(pField, eField) {
    var pResult = isValidKFUPMPhone(pField);
    var eResult = isValidEmail(eField);
    // display an alert when either of the above is false
    return true;
}
//-->
</script>
</head>
<body>
<h1>Validating Phone and E-mail:</h1>
<form name="idValid" onSubmit="return
  isReady(this.phone,this.email);" method="post" action="">
Phone : <input type="text" id="phone" /> <br />
E-Mail: <input type="text" id="email" /> <br />
<input type="submit" value="Submit" />
<input type="reset" value="Reset" />
</form>
</body>
</html>
```


Cont.

```
<script language="JavaScript">
<!--
function isValidKFUPMPhone(elm){
    var pattern = /0\d-?\d{3}-?\d{4}/;
    if(pattern.test(elm.value))
        return true;
    else return false;
}
function isValidEmail(elm){
    var pattern = /\w+\@\w+(.\w+)+/
    if(pattern.test(elm.value))
        return true;
    else return false;
}
// continued …
```

User-Defined Objects

- You can create complex data structures by creating your own objects
  - JavaScript allows that although it is not a full-featured OO language
- As usual, objects are created with a **constructor** function

```
function Employee(IDValue, NameValue, PayRateValue) {
    this.ID = IDValue;
    this.Name = NameValue;
    this.PayRate = PayRateValue.toFixed(2);
    this.Hours = 0;
    this.Pay = 0;
}
```

- JavaScript's constructors are like its other functions
- Such a method can be viewed as a *class* definition, providing the model for creating objects

Creating an Array of Objects

```
<script type="text/javascript">
var EmployeeDB = new Array();
function Employee(IDValue, NameValue, PayRateValue){
  this.ID = IDValue;
  this.Name = NameValue;
  this.PayRate = PayRateValue.toFixed(2);
  this.Hours = 0;
  this.Pay = 0;
}
function AddEmployees(){
  EmployeeDB[EmployeeDB.length] = new Employee("11111", "A. Katebah", 10.00);
  EmployeeDB[EmployeeDB.length] = new Employee("22222", "H. Al-Helal", 15.00);
  EmployeeDB[EmployeeDB.length] = new Employee("33333", "M. Araman", 20.00);
  EmployeeDB[EmployeeDB.length] = new Employee("44444", "F. Nabulsi", 25.00);
  EmployeeDB[EmployeeDB.length] = new Employee("55555", "Y. Al-Amer", 30.00);
}
</script>
```

Adding Methods to User Defined Objects

- Suppose you defined methods `ShowRecord()` and `ComputePay()` (with or without parameters)
- You add them to your object as follows

```
function Employee(IDValue, NameValue, PayRateValue){
    this.ID = IDValue;
    this.Name = NameValue;
    this.PayRate = PayRateValue.toFixed(2);
    this.Hours = 0;
    this.Pay = 0;
    this.ShowRecord = ShowRecord;
    this.ComputePay = ComputePay;
}
```


Using Your Object's Methods

```
function ShowRecord() {
    // code not shown
    return s;
}
function ComputePay(hours) {
    this.Hours = document.getElementById(hours).value;
    this.Pay = this.PayRate * this.Hours;
    this.Pay = this.Pay.toFixed(2);
}
function ShowEmployees() {
  var OutString = ""
  // code not shown
  for (i=0; i < EmployeeDB.length; i++) {
     OutString += EmployeeDB[i].ShowRecord();
  }
 // code not shown
}
function EnterHours() {
  for (i=0; i<EmployeeDB.length; i++) {
    if (EmployeeDB[i].ID == document.getElementById("EmployeeID").value) {
      EmployeeDB[i].ComputePay("EmployeeHours");
      break;
    }
  }
// code not shown
}
```

Debugging JavaScript

- If you mess up on the syntax you will get a Javascript Error
  - Netscape
    - You will see a notification of an error on the status bar in the bottom left corner
    - You type "javascript:" in the URL field to pinpoint the error
  - Internet Explorer
    - By default a tiny little Javascript error message appears at the bottom left corner of the browser in yellow. Usually you won't see it.
    - Can be explicitly disabled under Tools/Internet Options
    - Recommend under Tools/Internet Options/Advanced/Browsing to uncheck "Disable Script Debugging" and to check "Display a Notification about every script error" while doing development

Fixing JavaScript Errors

- If possible use the debugging tool to locate the line containing the error
- Errors can be hard to find and fix
  - "code a little, test a little" strategy
- Often errors are due to things that are easy to overlook, like not closing a quote

Q & A

References

- Some useful links with examples and other resources:
  - *Internet and World Wide Web How to Program*, 4/e, H. M. Deitel, P. J. Deitel, and A. B. Goldberg, Pearson Education Inc., 2008. Chapters 6-13.
  - *Web Development and Design Foundations with XHTML*, 4/e, Pearson Education Inc. 2009. Chapter 14.
  - http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.3/guide/intro.html
  - http://msconline.maconstate.edu/tutorials/JSDHTML/default.htm
  - http://www.javascript.com
  - Beginning JavaScript Tutorials http://www.pageresource.com/jscript/index.html
  - JavaScript Tutorial for the Total Non-Programmer http://www.webteacher.com/javascript/
  - More Beginning JavaScript Tutorials http://echoecho.com/javascript.htm
  - Core JavaScript 1.5 Reference Manual http://www.webreference.com/javascript/reference/core_ref
  - The JavaScript Source http://javascript.internet.com
- http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.3/guide/intro.html
- http://msconline.maconstate.edu/tutorials/JSDHTML/default.htm
- http://www.javascript.com
- http://www.elated.com/articles/form-validation-with-javascript/
- http://developer.apple.com/internet/webcontent/validation.html