Week 8: Form Handling

E-Mail Validation – p. 184, Script 8.10:

```
function validEmail(email) {
invalidChars = " /:,;"
```

// We must first define a function, **validEmail(email)**, to which the contents of the email field are passed.

// We then need to define a variable, **invalidChars**, that contains the five most likely invalid characters in an email address: blank space, slash, colon, comma and a semicolon. These will be tested later in the script.

// After we define our function we will ask some "**if**" statements

```
if (email == "") {
return false
}
```

// "**If**" email address field is equal to "**empty**" return false

// The next step is to search thru the email address for the invalid characters we defined earlier by creating an **initialization loop**

```
for (i=0; i<invalidChars.length; i++) {
badChar = invalidChars.charAt(i)
if (email.indexOf(badChar,0) > -1) {
return false
    }
}
```

// We need to use a "**for**" statement that starts an initialization loop that contains three steps to scan through the "**invalidChars**" string

// (1) The initialization step: "i=0" is where we define the initialization variable and set it to zero. In this case a blank character is at the beginning of the string value of the invalidChars variable that we will set the loop to look through.

// (2) The limiting step: "i<invalidChars.length" where we define when the loop should stop testing. In this case the length property of the characters in the string value of the invalidChars variable is the end.

// (3) The increment step: "i++" where we say by how much to increase the loop counter on each pass thru the loop.

// The **length** property equals the number of characters in the string

// After our initialization loop searches for invalid characters in our email address, the **badChar** variable saves the position of the invalid character of the **invalidChars** string.

// The "**charAt()**" method extracts (and saves) the character at a given position from a string.

// The "**indexOf()**" method searches the string for a character or substring.

```
// If the result of the loop finding an invalid character is at a
position of greater than "-1", when email.indexOf() looks for the
position of a badChar in the string, it will return a false result.


// Next, we need to check on the positions and number of at signs and
periods and make sure they are in the right place.


atPos = email.indexOf("@",1)
if (atPos == -1) {
return false
}


// The variable "atPos" defines where the position of our at sign
should be
// By using the method "indexOf()" we define that the position of our
at sign must be at least in the second position and if the position
of the at sign is equal to –1 we get a return of false


if (email.indexOf("@",atPos+1) > -1) {
return false
}


// We also must check to see that there is only one at sign by
checking to see if there are any more at signs after the first one it
found and if so we get a return of false


// Next, we need to do a similar test with the position of the period
making sure that there is at least one period after the at sign with
at least two characters after the period


periodPos = email.indexOf(".",atPos)
if (periodPos == -1) {
return false
}


// Define the variable "periodPos" to check and see if there is at
least one period after the at sign by using the method "indexOf()"
// "If" the position of our period (after the at sign) is equal to "-
1" we get a return of false


if (periodPos+3 > email.length)      {
   return false
}


// Make sure that there are at least two characters after our period
by asking "if" the position of the last period plus three characters
is greater than the length of our email address return false


return true
}
```

```
// If at this point your script has made it through all of these
tests without a any returns of false, you will get a return of true
when the form is submitted meaning that a valid email address has
been entered


// In order for the above tests to work a function needs to be
written that will be called from the form's submit button



function submitIt() {
if (!validEmail(carForm.emailAddr.value)) {
   alert("Invalid email address")
   carForm.emailAddr.focus()
   carForm.emailAddr.select()
   return false
}

return true
   }
```

// The function **submitIt()** is written to ask "**if**" the function **validEmail(email)** returns true then submit the form

// Notice that "**email**" is equal to "**carForm.emailAddr.value**"

// By using the "**!**" operator you ask "**if**" the function **validEmail(email)** returns false then **alert** the user, activate the object and select the field

```
<form onsubmit="return submitIt()" action="someAction.cgi"
name="carForm">
<table>
<tr>
<td>Email Address:</td>
<td><input name="emailAddr" type="text" size="30" /></td>
<td><input type="submit" value="Submit" /></td>
</tr>
</table>
</form>
```

// By clicking the **submit** button the user triggers the function "**submitIt()**" via the "**onsubmit**" user event thus calling the function "**validEmail(email)**"