

## COMP1536 Web Development – Lab 7

### Objective

Use JavaScript to add client-side validation to the data entry form developed in Lab 5

(Optional) Use the php script provided to email form inputs to a designated recipient.

### Evaluation

Complete this lab assignment before the close of your D2L Dropbox. W3C validation is required for all HTML/CSS files. Include both HTML and CSS validation logos and links. Submit both HTML and CSS files to your **D2L Dropbox**, and include the URL link as a Quicklink in the Comments box.

# Tasks

## 1. Id input controls

The first thing in form validation using Javascript is to make sure that there is a unique **id** attribute for each input control. Go through your form and set an **id** for each input control using *Reverse Hungarian notation* for each **id** (see screenshot below). E.g. for a **text** input field for last name, id the input **txtLastName**.

## 2. Creating a helper function called '\$'

The screenshot shows a web form with the following controls and IDs:

- Which position are you applying for? \*** (Dropdown menu) ID: `selPosition`
- Passed Comp 1536? \*** (Radio buttons)
  - Yes ID: `radPassedWeb_0`
  - No ID: `radPassedWeb_1`
- Earliest start date? \*** (Date input) ID: `txtStartDate_0,1,2`
- Portfolio URL** (Text input) ID: `txtPortfolio`
- Attach resume** (File input) ID: `filResume`
- Name \*** (Two text inputs)
  - First ID: `txtFirstName`
  - Last ID: `txtLastName`
- Email Address \*** (Text input) ID: `txtEmail`
- Phone \*** (Three text inputs) ID: `txtPhone_0,1,2`
- Any additional information for us** (Text area) ID: `taComments`

A Submit button is at the bottom.

Not part of the User Interface. These are **ids** to be used in your program.

One of the common ways to reference input controls in XHTML forms is to use the **getElementById()** method of the **document** object. Since you are going to be referencing a lot of form elements, let's first create a helper function named **\$**.

Add the following function to the **head** section:

```
function $(id){
    var element = document.getElementById(id);
    if( element == null )
        alert( 'Programmer error: ` + id + ` does not exist.' );
    return element;
}
```

Now, you can call `$('txtLastName')` instead of `document.getElementById('txtLastName')`.

### 3. Validate drop down list using *onchange* event

- a. When the user makes a selection on the drop down list, call Javascript functions to validate the user selection. The way to do this is to add the `onchange` event handler to the `select` element. The property is `onchange` and the value is a validation function call. Test to make sure it works.

```
<select ...  
    onchange="warnPositionInvalid('selPosition')">  
    ...  
</select>
```

- b. For validation, separate out the 'test' and 'warn' functions as per the code below. This will make it much easier for the next lab that involves modifying the validation strategies outlined in this lab. Note that for the `select` input type, the `selectedIndex` property returns an integer representing the user selected option, with 0 representing the first option.

```
function testPositionValid(id){  
    return ($(id).selectedIndex != 0);  
}  
  
function warnPositionInvalid(id){  
    if( !testPositionValid(id) )  
        alert( "You must select a position" );  
}
```

- c. Now replace `onchange` with `onblur`. Test your web form – *what is the difference between these two events?*

### 4. Validating the remaining inputs

For each of the remaining form inputs in the table below that requires validation, do the following:

- Add an appropriate event handler to the input element.
- Write a pair of *test* function and a *warn* function that uses the *test* function. Refer to the information on properties for input types provided in the appendix.
- Alert the user of any invalid inputs.
- Test your form validation thoroughly.

Input	Validation
Passed Comp 1536	If no, warn them that they will not be considered if they have not passed this course. Both inputs (Yes and No) should call the same validation function
Earliest start date	If any of the three inputs are empty, then provide no warnings (why?). If all three inputs have values, ensure the values form a valid date: 1 <= day <= 31, 1 <= month <= 12, 2013 <= year <= 2999. Note all of the date subfields should call the same validation function.
Portfolio URL	No validation required
Attach resume	If the field is not blank, verify that the last three characters in the filename are either <b>doc</b> , <b>docx</b> or <b>pdf</b> . Hint: given a string <b>str</b> , the last three characters can be extracted with <b>str.substring(str.length-3)</b>
Name	First name and last name must not be blank. Hint: given a string <b>str</b> ,

	<b>str.length</b> returns the length of the string.
Email	E-mail must not be blank and has suffix “.com”, “.ca”, or “.org”
Phone	If any of the three are blank, provide no warnings. Otherwise verify that the first two components are 3-digit values ( $100 \leq num \leq 999$ ) and the last value is a four digit value ( $1000 \leq num \leq 9999$ )
Comments	No validation required

## 5. (Optional) Email form inputs to recipient

Add formmail.php as the action to your form. In order to get this form working, do the following:

- Modify line 260 of formmail.php to specify the domain name of the email recipient. (E.g. if the recipient domain name is outlook.com, the line should read:

```
$TARGET_EMAIL = array(EMAIL_NAME."@outlook\com$");
```

- Add the following hidden fields to your form and modify the value attributes:

```
<input type="hidden" name="recipients" value="you@your-host-name.com" />
<input type="hidden" name="subject" value="Sample FormMail Testing" />
<input type="hidden" name="good_url" value="thankyou.html" />
```

- Create a thankyou.html page after the form has been submitted.

### Appendix:

Here are the common properties for input types:

Input type	Key property	Description
Select	selectedIndex	Integer representing the user selected option (0 == first option)
Textbox	value	String containing user entered text
File	value	String containing filename provided by user
Textarea	value	String containing user entered text
Radio buttons	checked	Boolean value indicating whether the button is checked (true) or not (false)

## Marking scheme

Item	Marks
All user input controls must be validated with Javascript functions	7
Html,CSS files are well structured, indented and commented	2

Quicklink inserted into D2L Comments box	1
TOTAL:	10