|  |  |
| --- | --- |
| Usability Engineering | CS2511 Practical Sessions |

# Introductory/Revision Exercise (i)

## Client-side scripting and Form Elements

The aim of this exercise is to explore the use of client-side scripts to process data entered via form elements on web-pages.

* Using a text-editor, create a web-page that includes a form, e.g.,:

<form name="theForm">

</form>

Note that the form should have a name, but don't bother specifying an action or method since the form won't be submitted.

* Within the form, create a text-area, e.g.:

<textarea name="txtBox">Some text...</textarea>

* Outside the form-tags, create a button, e.g.:

<button onclick="procForm()">Process Form</button>

View the page in a browser and check that the text-area and button display correctly.

* In the head section of the HTML document, create a script, e.g.:

<script type="text/javascript">

</script>

* Within the script tags, write a function that is called when the button is clicked. It should have the same name as that specified in the onclick attribute of the button, e.g.:

function procForm() {

}

* Within the function, write a line of code that recovers the text within the text-area and displays it in a dialog-box. You can access the properties of form-elements in the following way:

document.<form-name>.<element-name>.property

e.g.:

document.theForm.txtBox.value

(value is a property which represents the text in a text-area. It can be read or written. Other properties of a text-area, such as its size, color, visibility, etc., can be accessed in the same way as the value.)

To display the recovered text, use an alert dialog-box, e.g.:

alert(data);

Where data is a quoted string or variable to be displayed. In this case, it should be the reference to the text in the text-area.

View the page in a browser. Clicking the button should cause a dialog-box to appear, containing the contents of the text-area.

Add a second text-area to the page. Make sure it has a unique name. Re-write the function so that it copies the contents of the first text-area into the second.

Add a selection-box to the page. Selection-boxes are created as follows:

<select name="selBox">
    <option>Option 1</option>
    <option>Option 2</option>
    <option>...</option>
    <option>Option n</option>
</select>

View the page in a browser and check that the selection-box displays correctly.

Write a function which is called whenever the selection in the selection-box is changed, and which displays (in a dialog-box) the new selection.

* You can call the function by adding an onchange event-handler to the selection-box. This works in much the same way as the onclick event-handler used with the button. Make sure the function-name specified in the event-handler matches the funtion you wish to call in the script.
* The properties of selection-boxes can be accessed in the same way as those of text-areas. The most important property of a selection-box is the selectedIndex, an integer which indicates which of the options has been selected. Obtain this property and display it in a dialog-box whenever the sleecdtion is changed. Check that the displayed value is what you expect (e.g., '0' if the first option is selected, '1' if the second option has been selected, and so on).
* The selectedIndex indicates which of the options has been selected, but not what it is. Modify the function so that it reports the text of the selected option. The options within a selection-box are organised as an array, and the text within each option is held in a property called text. Therefore you can access the text of a particular option (either to read it or write it) in the following way:

document.selBox.options[i].text

Where i is the selectedIndex of the required option.