Lab Practical - Javascript Basics

Create a page that uses document.write()

Aim: To create a tiny page that includes some elements, then add a <script> block to include some additional content using document.write()

Calling Javascript functions

Create a page that contains two buttons and arrange for the button to call a javascript function `popup()` that you've written.

The function `popup(x)` is really simple - it just displays the parameter *x* in an alert box.

Write this in two ways:

1. with the function inside a <script> … </script> block
2. with the function code inside an external file.

Getting content from a text box

Create a page with a text box, and a submit button

- the text box has an id='reply' tag
- when the submit button is pressed, display the content of the text box in an 'alert()' popup.

To get the value from the text box, you'll need to use the *getElementById(id-Tag)* function.

Once you've got access to the right element, you can extract content:

e.g.

```
v = document.getElementById('reply').value
```

Or update the contents of an existing element:

```
document.getElementById('idTag').innerHTML = 'new value'
```

where 'idTag' is the id of some element (e.g. a <p>)

if statements

Modify this example so it:

1. only displays a popup if the textbox value is contains 'fred'

1. has a counter so that it counts the number of times the popup occurs and displays that along with the textbox value
2. uses a conditional expression rather than an *if-statement* to convert *fred* to *FRED*

for loops

1. Add a paragraph heading with the id of 'output' and display all the numbers from 1 to 10 using a for loop.
2. Create an text box for the upper limit and display values from 1 to this limit
    - using the function **parseInt(s)** to convert the string from the textbox.

Checking the Numeric Conversion:

- Only run the for loop if the result from *parseInt()* is a number.

If parseInt() returns **NaN**, display an alert box with "Not a valid number: " and then the textbox contents.

Arrays

Make a page that has a textbox and a two buttons:

- *Add* - which adds the textbox contents to the array
- *Show* - which output all of the stored items in the array into a <h3> tag.

Then add these buttons:

- *Clear* - which deletes all the elements from the array
- *Delete-Last* - which deletes the most recently added element

Try playing with change-document-font to do other things

```
<h4>Change Body Font</h4>

<input type=submit value='Body Font is Arial  ' onClick="setFont('Arial'  )"> <br>
<input type=submit value='Body Font is Georgia' onClick="setFont('Georgia')"> <br>
<input type=submit value='Body Font is Times  ' onClick="setFont('Times'  )"> <br>
<input type=submit value='Revert Body Font    ' onClick="revertFont()"      > <br>
<input type=submit value='Show Changes        ' onClick="showChanges()"     >

<p><b>Changes</b></p>
<p id = 'changeList'> None yet </p>

<script>
originalFont=null  // Global - place to save original
fontChanges = Array();

function setFont(whichFont){
   if (originalFont == null) {
      originalFont = document.body.style.fontFamily;
   }
   document.body.style.fontFamily = whichFont
   fontChanges.push(whichFont)
}

function showChanges(){
   if (fontChanges.length == 0)
     alert("No changes")
   else {
      s = ''
      for (i=0; i<fontChanges.length; i++) {
        s = s + fontChanges[i] + '<br>'
      document.getElementById('changeList').innerHTML = s
      }
   }
}
```

Course: 158.256 Web Dev

Validate