Creating dynamic web pages – Scripts

A client-side script is a program that may accompany an HTML document or be embedded directly in it. The program executes on the client's machine when the document loads, or at the time some other event occurs.

This program code, written in a scripting language understandable by your web browser, can be used to dynamically create and/or modify web page content. This program code must be enclosed in `<script>` and `</script>` tags and the language must be identified by the value of a type attribute in the `<script>` tag. We will use Visual Basic as a scripting language. A note of caution here: Microsoft's Internet Explorer is the ONLY web browser that understands this scripting language, so it isn't a good idea to use this particular language to create dynamic web content for the general public. A section of VB script embedded in a web page has the following appearance.

```html
<script type="text/vbscript">
<!--
    ' your VB statements
    ' go here
    ,
    ' you can define functions or subroutines (which must be called in order to be executed)
    ,
    ' you can also just write code to be executed as it is read
    ,-->
</script>
```

Browsers or other user agents that don't recognize the `<script>` element would likely render that element's contents as text. Some scripting engines, like VBScript, allow the script statements to be enclosed in an HTML comment as indicated above. This way, user agents that don't recognize the `<script>` tag will ignore the comment (and the script) while smart scripting engines will understand that the script in comments should be executed.

Scripts that are executed when a document is loaded may be able to create or modify the document's contents dynamically. The "document.write" statement in the HTML object model is supported by several scripting languages, including VBScript. All scripts are evaluated in order as the document is loaded by your browser. If a script generates HTML content as its output, that content is inserted into the document in the place where the `<script>` tag occurred.

Here is an example of the use of the document.write statement.

```html
<html>
<head><title>Dynamically Generated Content</title></head>
<body>
<script type="text.vbscript">
<!--
    for var=0 to 20
    document.write(“Line “ & var & “ of dynamically generated content”)
    next
    ,-->
</script>
</body></html>
```
A function or subroutine can also be enclosed in `<script>` and `</script>` tags. Then you must figure out a way to call that function or subroutine in order to get it to execute. This is done using the events that are associated with the different HTML elements.

**Creating Dynamic HTML pages – Events and attributes**

Each of the tags in your HTML documents has attributes. There are several standard attributes which every tag has. These are

- **class** – used for applying style information, we won't worry about this attribute now
- **id** – a unique name, which can be referenced by scripting languages like vbscript
- **title** – text, used as a tool-tip when the mouse hovers over the tagged content
- **style** – in-line style information, we won't worry about this attribute now

HTML elements also have event attributes. Most HTML tags can *respond* to a number of different events if the corresponding event attribute has been *assigned an appropriate value*. Event attributes include

- **onload** – only applicable to the `<body>` tag, occurs when page is loaded
- **onunload** – only applicable to the `<body>` tag, occurs when page is unloaded
- **onclick** - occurs when the pointing device button is clicked over an element
- **ondblclick** - occurs when the pointing device button is double clicked over an element
- **onmousedown** - occurs when the pointing device button is pressed over an element
- **onmouseup** - occurs when the pointing device button is released over an element
- **onmouseover** - occurs when the pointing device is moved onto an element
- **onmouseout** - occurs when the pointing device is moved away from an element
- **onmousemove** – occurs when the pointing device is moved within
- **onkeydown** - occurs when a key is pressed down over an element
- **onkeyup** - occurs when a key is released over an element
- **onkeypress** - occurs when a key is pressed and released over an element

**Creating Dynamic HTML pages – Controls and their attributes**

HTML defines the following control types for designing user interfaces:

- **button** (submit and reset buttons are only for use with forms, the ordinary push button can have client-side actions associated with its even attributes)
  ```html
  <button type="button" name="execute" onclick="doit()">
  <input name="howHigh" type="button" onclick="jump(12)">
  ```

- **checkbox** (on/off switches)
  ```html
  <input name="PumpkinPie" type="checkbox">Pumpkin Pie<br>
  ```

- **radio button** (mutually exclusive choices)
  ```html
  <input name="sex" value="male" type="radio">Male<br>
  <input name="sex" value="female" type="radio">Female<br>
  ```

- **menu** (requires both beginning and ending tags)
  ```html
  <select name="whatToDo" onchange="update()">
  <option onclick="scram()">Run</option>
  <option onclick="duck()">Hide</option>
  <option onclick="yield()">Surrender</option>
  </select>
  ```
- **text input** (textarea is multi-line, input control can create a single-line input area)
  `<input type="text" value="initial value">`
  `<textarea rows="5" cols="40">`

- and a few others

Additional events associated with `<a>` anchor tags and with the label, input, select, textarea and button controls are

- **onfocus** – occurs when an element receives focus either by the pointing device or by tabbing navigation
- **onblur** – occurs when an element loses focus either by the pointing device or by tabbing navigation

Events associated with input, select and textarea controls are

- **onselect** – occurs when a user selects some text in a text field (input and textarea controls only)
- **onchange** – occurs when a control loses the input focus and its value has been modified since gaining focus

Events associated with an HTML form element are

- **onsubmit** - occurs when a form is submitted
- **onreset** - occurs when a form is reset

---

**Creating Dynamic HTML pages – Forms**

Finally, we will briefly mention the topic of forms and the `<form> </form>` tags. An html form is simply a container for some html controls. The different types of controls that can be placed in an html form are the ones we've mentioned before

- button
- textarea
- label
- select
- input

A form normally contains one or two special buttons labeled as Submit and Reset buttons. These are use to clear the form (Reset) and to send the contents of the form to a server application at a remote web site (Submit). In addition, the events associated with the other controls can be used to provide client-side functions. We won't use forms in our exercises because of the complexity of writing a server-side application to process the form. All the same controls (except for Reset and Submit) can be used outside of the form tags to design a client-side user interface.