Lecture 04 AJAX

EGCI427

This set of slides and examples are modified from "Programming the World Wide Web" By Robert W. Sebesta, 4th edition.

JavaScript Execution Environment

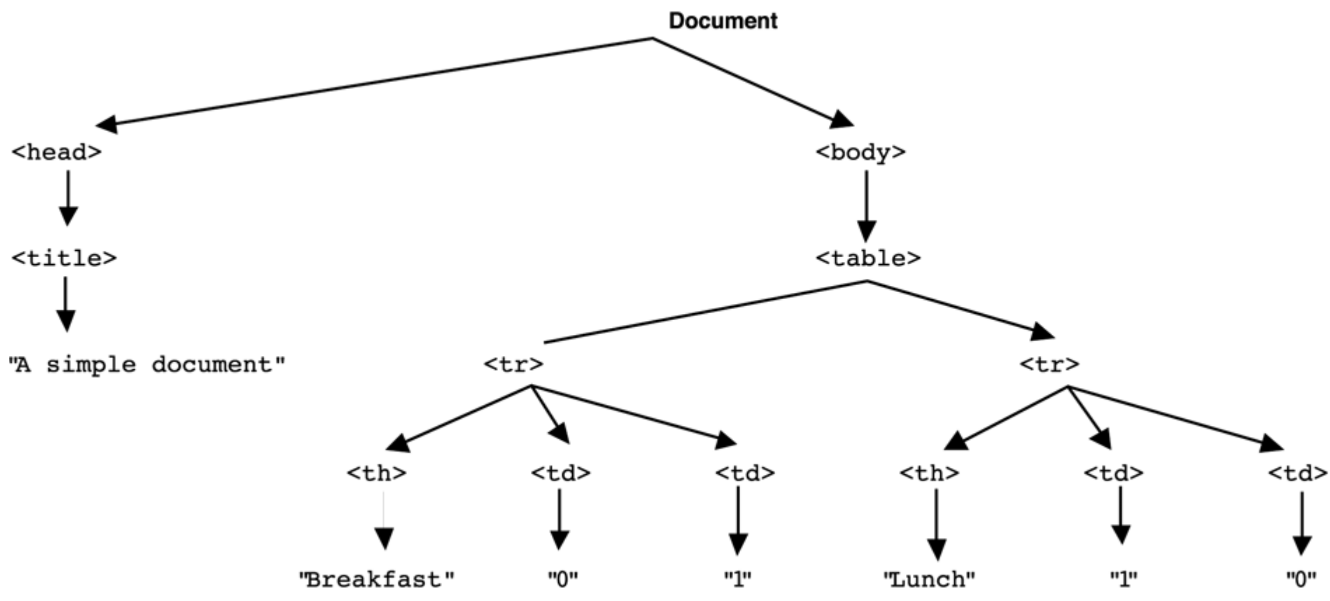
- JavaScript executing in a browser
- The Window object represents the window displaying a document
  - All properties of the `window` object are visible to all scripts
  - Global variables are properties of the Window object
  - There can be more than one Window object
    - Global variables depend on which Window is the context
- The Document object represents the document displayed
  - The document property of Window
  - Property arrays for forms, links, anchors, …
- The frames property of Window

Document Object Model (DOM)

- To provide a specification programs or scripts that deal with XHTML portable among various browser
- DOM Levels
  - DOM 0: informal, early browsers
  - DOM 1: XHMTL/XML structure
  - DOM 2: event model, style interface, traversal
  - DOM 3: content model, validation
- DOM specifications describe an abstract model of a document
  - Application Programming Interface (API)
  - Interfaces describe methods and properties
  - The interfaces describe a tree structure
  - Different languages will bind the interfaces to specific implementations
    - The internal representation may not be tree-like
    - In JavaScript, data are represented as properties and operations as methods

Example

```
<html xmlns="http://www.w3.org/19/xhtml>
<head><title>A simple document</title></head>
<body><table>
<tr> <th>Breakfast</th>
      <td>0</td>
      <td>1</td>
</tr>
<tr> <th>Lunch</th>
      <td>1</td>
      <td>0</td>
</tr>
</table> </body> </html>
```



Example

- The HTML document on the previous slide is shown as a conceptual tree
- Nodes of the tree will be JavaScript objects
- Attributes of elements become named properties of element node objects
  - <input type="text" name="address">
  - The object representing this node will have two properties
    - type property will have value "text"
    - name property will have value "address"

Element Access in JavaScript

- Elements in XHTML document correspond to objects in JavaScript
- Objects can be addressed in several ways:
  - `forms` and `elements` array defined in DOM 0
    - Individual elements are specified by index
    - The index may change when the form changes
  - Using the name attributes for form and form elements
    - A name on the form element causes validation problems
    - Names are required on form elements providing data to the server
  - Using getElementById with id attributes
    - id attribute value must be unique for an element

Using forms array

- Consider this simple form:

```
<form action = "">
     <input type = "button"  name =
"pushMe">
  </form>
```

- The input element can be referenced as

```
document.forms[0].element[0]
```

Using name Attributes

- All elements from the reference element up to, but not including, the body must have a name attribute
- This violates XHTML standards in some cases
- Example

```
<form name = "myForm"  action = "">
   <input type = "button"  name = "pushMe">
</form>
```

- Referencing the input

```
document.myForm.pushMe
```

- XHTML1.1 does not allow `name` attribute in the form element
  - Only validation problem but causes no difficulty for browsers

Using id Attribute

- Set the id attribute of the input element

```
<form action = “”>
   <input type=“button”  id=“turnItOn”>
</form>
```

- Then use getElementById

```
document.getElementById(“turnItOn”)
```

Example of XMLHttp

XMLHttp

- innerHTML
  - Update text based on user input
  - document.getElementById(‘*elementID*').innerHTML = ‘Text';
- XMLHttpRequest
  - The XMLHttpRequest object is used to exchange data with a server behind the scenes
  - xmlhttp=new XMLHttpRequest();
- responseText
  - Returns the response data as a string
  - xmlhttp.responseText

XMLHttp

- xmlhttp.readyState==4
  - 0 Uninitialized - open() has not been called yet.
  - 4 Completed - Finished with all operations.
- xmlhttp.status==200 is OK

- open(*method,url,async*)
  - Specifies the type of request, the URL, and if the request should be handled asynchronously or not.

    method: the type of request: GET or POST
    url: the location of the file on the server
    async: true (asynchronous) or false (synchronous)

XMLHttp

- send(*string*)
  - Sends the request off to the server.

    string: Only used for POST requests

- xmlhttp.open("GET",“Text",true);
  xmlhttp.send();