JAVASCRIPT and the DOM

Les Carr
COMP3001
JavaScript and the DOM
Behavioral Layer

Web pages have 3 layers...

• **Structural/Content Layer** *(XHTML)*
  – The meat and potatoes

• **Presentational Layer** *(CSS)*
  – How things look; garnishing the meat and potatoes on a pretty plate

• **Behavioral Layer** *(JavaScript and DOM)*
  – How websites behave; the meat can jump off the plate if you want it to.
Client-side Languages

- User-agent (web browser) requests a web page
- JavaScript is executed on PC
- Can affect the Browser and the page itself
- Web page (with JavaScript) is sent to PC
Client-side

What kind of things can you do with JavaScript?

• Validating Form information,
  – i.e., making sure all the fields are complete before submitting data back to the server

• Modifying a web page based on Mouse Events.
  – Can turn a web page into a user interface with interactive buttons and controls
Server-side Languages

- User-agent (web browser) requests a web page
- User never sees the PHP, only the output
- Cannot affect the browser or client PC
- Web page (with PHP Output) sent to PC
JavaScript

• Java Functions definitions are embedded in the `<head>`

• Function calls are placed in the `<body>`

```html
<html>
<head>
    <script language="javascript">
        function myfun() {
            do something;
        }
    </script>
</head>
<body>
    <script language="javascript">
        myfun();
    </script>
</body>
</html>
```
DOM Scripting

Key Topics:
• Event Handling
• The Browser Object
• Document Object Model
  – the document structure
Event Handling

- JavaScript code can be initiated by browser events
  - HTML 4.0 supports lots of events.
  - onclick, onchange, onmousedown, onmousemove, etc.
Browser Events

- **onblur** – an element loses focus, i.e., click on a text box, but then you click on something else; the text box is blurred
- **onchange** – contents of an element is changed, i.e., changing the selection in a drop down menu
- **onfocus** – an element is clicked or selected
- **onload** – when the web page is initially loaded
- **onsubmit** – when a form’s submit button is clicked
More Browser Events

- **onkeydown** – immediately when a key is pressed down
- **onkeypress** – if the key is held down, i.e., not immediately released
- **onkeyup** – immediately when a key is released.
  - Sometimes, you want something to happen when the key goes down vs. goes up
  - Sometimes, you want to detect a long key press
    - SHIFT, CTRL, or ALT
Even More Browser Events

- **onmousedown** – a mouse button is pressed down
- **onmouseup** – a mouse button is released
- **onmousemove** – a mouse is moved
- **onmouseout** – mouse is moved off an element (blur)*
- **onmouseover** – mouse is moved on an element (focus, hover)*

* Used for hover effects.
Example

<html>
<body>
<h1>Example Javascript Event Handler</h1>
<p>Here is some text with a
   <span onClick="alert('Do not click here')">sensitive patch</span> in it</p>
in it
</body>
</html>
Example 2

<html>
<head><script language="JavaScript">
function log(s){window.status=s}
</script></head>
<body>
<h1>Example Javascript Event Handler</h1>
<p>Here is some text with a
   <span onMouseOver="log('Do not click here')"
       onMouseOut="log(' ')" > sensitive patch</span>
in it</p>
</body>
</html>
Example 3

<html>
<head>
<script language="JavaScript">
function log(s){window.status=s}
</script></head>
<body>
<h1>Example Javascript Event Handler</h1>
<p>Here is some text with an
 <a href="javascript:log('Flip off out of here')"> insensitive patch</a> in it</p>
</body>
</html>
DOM Scripting

First a summary:
1. JavaScript can be initiated by browser events.
2. JavaScript can access and manipulate the browser object.

What’s Next
• JavaScript can access the document structure.
<p>This is a paragraph linking to <a href="http://www.soton.ac.uk">Southampton</a>.</p>
DOM Script Example

```html
<body>
  <div id="menu">
    <h1>Main Menu</h1>
  </div>

  <div id="content">
    <h1>Page Title</h1>
    <p>Blaa blaa blaa.</p>
  </div>
</body>
```
DOM Script Example

```html
<html>
<div id="menu">
<h1>Main Menu</h1>
</div>

<div id="content">
<h1>Page Title</h1>
<p>Blaa blaa blaa.</p>
</div>

<script language="JavaScript">
var contentdiv = document.getElementById('content');
var pagetitle = contentdiv.getElementsByTagName('h1')[0];

pagetitle.setAttribute("style","color:red");
pagetitle.firstChild.nodeValue="The Red Page Title";
</script></html>
```
DOM Script Example

```javascript
var themenu = document.getElementById('menu');
var thebody = menu.parent;
var thecontent = menu.nextSibling;
var contentnodes = thecontent.childNodes;
var theh1 = contentnodes[0];
var firstparagraph = contentnodes[1];
```

Diagram:
- **body**
  - **div id="menu"**
  - **h1**
    - **Main Menu**
  - **div id="content"**
    - **h1**
    - **p**
      - **Page Title**
      - **Blaa blaa blaa.**
DOM Scripting Functions

Modifying Structure
• insertBefore()
• appendChild()
• replaceChild()
• removeChild()
• cloneNode()

Creating Elements
• createElement()
• createTextNode()

Modifying Attributes
• getAttribute() / setAttribute()
DOM and Forms

- Every form in a page is held in an array
  - `document.forms[0]` is the first form

- Every component (input, select or textarea element) is held in a subarray
  - `document.forms[0].elements[0]` is the first field

```html
<form name="personal">
  <input type="text" name="name">
  <input type="text" name="address">
  <input type="text" name="city">
</form>
```

- `either document.forms[0].elements[1]` or `document.forms["personal"].elements["address"]`
- `or document.personal.address`
DOM and Forms

- Every component of the form has a value
  - `document.personal.address.value`

- The value can be used in expressions or stored in an assignment statement

- Specific components have specific methods or properties
  - A menu (ie a select) has property ‘selectedIndex’
  - A checkbox has property ‘checked’

- An `onsubmit` event handler can check its form’s components and halt the submission by returning `false`
DOM and Forms

```html
<form name="personal" onSubmit="validate()">
    <input type="text" name="name"/>
    <input type="text" name="address"/>
    <input type="text" name="city"/>
    <input type="submit" value="Submit!"/>
</form>

<script lang="JavaScript">
function validate(){
    if(document.personal.name.value.length==0){
        alert("Missing name");
        return false;
    }
    return true;
}
</script>
```