Ajax Technology in Web Programming

Sergio Luján Mora

API for accessing and manipulating HTML documents

DOM
Introduction

• **Document Object Model (DOM)**
  – Allows accessing HTML elements in a document → It isn’t a programming language
• **Browser Object Model (BOM)**
  – *Navigator objects*
  – Allows accessing browser’s components
• There are some incompatibilities between different browsers
Introduction

• Benefits:
  – Allows client-side processing without the need of web server processing

DOM

• The Document Object Model is an API for HTML and XML documents
• W3C Specification
• It does two things for web developers:
  – it provides a structural representation of the document (tree structure), and
  – it defines the way that that structure is to be accessed and manipulated from script, allowing you to get at the web page as a structured group of nodes
• Essentially, it connects web pages to scripts or programming languages
DOM

• Example:

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01">
<html>
<head>
<title>Ejemplo de DOM</title>
</head>
<body>
<!-- es un ejemplo un poco simple -->
<p style="color:red">Bienvenidos al <b>DOM</b></p>
</body>
</html>
```

DOM

• Nodes reflect content and structure of the document:
DOM

Example:

```html
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```
DOM

- DOM levels:
  - Level 0
  - Level 1
    - Document Object Model Level 1 (1/10/1998)
  - Level 2
    - Document Object Model Level 2 Core (13/11/2000)
    - Document Object Model Level 2 Views
    - Document Object Model Level 2 Events
    - Document Object Model Level 2 Style
    - Document Object Model Level 2 Traversal and Range
    - Document Object Model Level 2 HTML
  - Level 3
    - Document Object Model Level 3 Load and Save
    - Document Object Model Level 3 Validation

What does your user agent claim to support?

- [http://www.w3.org/2003/02/06-dom-support.html](http://www.w3.org/2003/02/06-dom-support.html)
<table>
<thead>
<tr>
<th>Feature</th>
<th>DOM Level 1</th>
<th>DOM Level 2</th>
<th>DOM Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core: basic methods (Level 1 and 2) and extensions for XML Namespaces (Level 2 only)</td>
<td>-</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>XML: extensions for XML 1.0</td>
<td>supported</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>HTML: extensions for HTML 4.0 (Level 1 and 2) and support of HTML 1.0 (Level 2 only)</td>
<td>supported</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>Views: used with the Level 2 CSS and UIEvents DOM modules</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>StyleSheets: association between a stylesheet and a document</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>CSS: extensions for cascading style sheet</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>CSS2: extensions for Cascading Style Sheets Level 2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Events: generic events system</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>UIEvents: basic user interface events</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MutationEvents: mouse device events</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MutationEvents: events for mutations in a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>HTMLEvents: HTML 4.01 events</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Range: selections to manipulate a range in a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Traversal: Alternative traversal methods of a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>LS: Loading a document into a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>LS-Async: Asynchronous loading of a document into a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Validation: Schema-oriented modification at a DOM tree</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
DOM Level 0

- The term "DOM Level 0" refers to a mix (not formally specified) of HTML document functionalities.
- Offered from Netscape Navigator version 3.0 and Microsoft Internet Explorer version 3.0.
- It is also a set of APIs which are not a part of any W3C DOM specification, but are implemented across several browsers → No standard.

DOM Level 0

- `innerHTML`:
  - Sets or retrieves the HTML between the start and end tags of the object.
  - For example, the entire contents of the document body can be deleted by:

```javascript
// Replaces body content with an empty string.
document.body.innerHTML = "";
```
DOM Level 0

- As there is no public specification for this property, implementations differ widely.
- It should never be used to write parts of a table—W3C DOM methods should be used for that—though it can be used to write an entire table or the contents of a cell.

---

DOM Level 0

- Example: text changes when cursor mouse is moved

```html
<p onmouseover="this.innerHTML='<b>Mouse out to change back.</b>'"
    onmouseout="this.innerHTML='<i>Mouse over again to change.</i>'">
<i>Mouse over this text to change it.</i>
</p>
```
DOM Level 1

- Two parts:
  - **Core**: a low-level set of fundamental interfaces that can represent any structured document, as well as defining extended interfaces for representing an XML document.
  - **HTML**: provides additional, higher-level interfaces that are used with the fundamental interfaces to provide a more convenient view of an HTML document.

DOM Level 1

- Every node is a type of `Node`, but there are different subtypes: `Document`, `DocumentType`, `Element`, `Text`, `Comment`, ...

- Attributes are nodes of type `Attr`, but they are not in the tree that represents the document (attribute value is retrieved from a method of the node)
DOM Level 1

• Every node has a set of properties that link to the “relatives”:
  - childNodes
  - firstChild
  - lastChild
  - parentNode
  - nextSibling
  - prevSibling

• Example:
  - A.firstChild = A1
  - A.lastChild = A3
  - A.childNodes.length = 3
  - A.childNodes[0] = A1
  - A.lastChild.firstChild = A3a
  - A3b.parentNode.parentNode = A
  - A1.nextSibling = A2
  - A3.prevSibling = A2
  - A3.nextSibling = null
DOM Level 1

- `getElementById("elementID")`
  ```javascript
  var element = document.getElementById("myTable");
  ```

- `getElementsByTagName("tagName")`:
  ```javascript
  var images = document.getElementsByTagName("img");
  ```
  The special value "*" matches all tags.

- `getAttribute("attrName")`:
  ```javascript
  var element = document.getElementById("myTable").getAttribute("width");
  ```

// HTML:
// <div id="d"><p>Content</p>
// <p>Further Elaborated</p>
// </div>

d = document.getElementById("d");
alert(d.innerHTML);
DOM Level 1

• createElement(“tagName”)
  - var element =
    document.createElement(“tbody”)

• appendChild(element)
  - element.appendChild(newChild);

• replaceChild(element, element)
  - element.replaceChild(newElement, 
oldElement);

• setAttribute(name, value)
  - document.getElementById(“myTable”).
    setAttribute(“width”, 300);

How to access a web form

• Three ways:
  - document.forms[n]
  - document.forms[“miForm”]
  - document.miForm

• General syntax for accessing a property
  of a control in a form:
  - document.nForm.nControl.propiedad
Ajax Technology in Web Programming

How to access a web form

• Checkbox:
  - checked
  - defaultChecked
  - value

• Radio button:
  - checked
  - defaultChecked
  - value

• Text field, password field and text area:
  - defaultValue
  - value

• Selection list:
  - length
  - options → selected, text, value
  - selectedIndex
BOM

- **Browser Object Model**
- Under specification process by W3C:
  - Window Object 1.0 W3C Working Draft 07 April 2006
- Objects:
  - window
  - history
  - location
  - navigator
- **Root object**: `window (it can be omitted)`
  
  ```javascript
  window.navigator.appName
  navigator.appName
  ```

Object window

- Main object
- Represents a window (body) or a frame (frameset)
- Properties (some are objects):
  - `document, frames, history, location`
  - `parent, self, top`
  - `status and defaultStatus`
Each frame is represented as a window object.

- `self` property retrieves a reference to the current window or frame.
- `parent` property retrieves the parent of the window in the object hierarchy.
- `top` property retrieves the topmost ancestor window, which is its own parent.
- If `self` is equal to `self.parent`, then the current window doesn’t have a parent, meaning it is the topmost window.
Object window

- Methods:
  - alert(mensaje)
  - clearInterval(intervalID)
  - clearTimeout(timerID)
  - confirm(mensaje) → Ok(true)/Cancel(false)
  - prompt(mensaje[, valorInicial])
  - close
  - open(URL, nombre[, características])
  - setInterval(expresión, milisegundos)
  - setTimeout(expresión, milisegundos)

<table>
<thead>
<tr>
<th>Parámetro</th>
<th>Tipo valor</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>toolbar</td>
<td>boolean</td>
<td>display a toolbar</td>
</tr>
<tr>
<td>location</td>
<td>boolean</td>
<td>display the location text box</td>
</tr>
<tr>
<td>directories</td>
<td>boolean</td>
<td>display the special link buttons</td>
</tr>
<tr>
<td>status</td>
<td>boolean</td>
<td>display a status bar</td>
</tr>
<tr>
<td>menubar</td>
<td>boolean</td>
<td>display the menus at the top of the window</td>
</tr>
<tr>
<td>scrollbars</td>
<td>boolean</td>
<td>display scrollbars if the document is larger than the window</td>
</tr>
<tr>
<td>resizable</td>
<td>boolean</td>
<td>allow the window to be resized</td>
</tr>
<tr>
<td>width</td>
<td>integer</td>
<td>the width of the window (in pixels)</td>
</tr>
<tr>
<td>height</td>
<td>integer</td>
<td>the height of the window (in pixels)</td>
</tr>
<tr>
<td>top</td>
<td>integer</td>
<td>the top position of the window (in pixels)</td>
</tr>
<tr>
<td>left</td>
<td>integer</td>
<td>the left position of the window (in pixels)</td>
</tr>
</tbody>
</table>
Ajax Technology in Web Programming

Object window

<html>
<head>
<title>Prueba de window.open</title>
<script type="text/javascript">
window.open("http://www.ua.es/", "ventana1", "width=200, height=200, toolbar=no, menubar=no, location=no, directories=no");
window.open("http://www.ua.es/", "ventana2", "width=300, height=300, location=yes");
window.open("http://www.ua.es/", "ventana3", "width=400, height=400, menubar=no, toolbar=yes, location=yes");
</script>
</head>
<body>
</body>
</html>
Object history

- Array of visited URLs in the current window
- Methods:
  - `back()`: previous visited page
  - `forward()`: following page
  - `go(n)`: go to page ‘n’ in the list of URLs

Object location

- Holds the URL of the current page shown in the window (or frame)
- Properties:
  - `host`, `hostname`, `href`, `pathname`, `port`, `protocol`, `search`, `hash`
- Methods:
  - `reload()`, `replace()`
Object navigator

• Information about the browser (version, operating system, etc.)
• Properties:
  – appCodeName, appName, appVersion, platform

```html
<html>
<head>
<title>Ejemplo de uso del objeto location</title>
</head>
<body>
<script type="text/javascript">
  document.writeln("navigator.appCodeName = " + navigator.appCodeName);
  document.writeln("<br />");
  document.writeln("navigator.appName = " + navigator.appName);
  document.writeln("<br />");
  document.writeln("navigator.appVersion = " + navigator.appVersion);
  document.writeln("<br />");
  document.writeln("navigator.language = " + navigator.language);
  document.writeln("<br />");
  document.writeln("navigator.mimeTypes = " + navigator.mimeTypes);
  document.writeln("<br />");
</script>
</body>
</html>
```
Ajax Technology in Web Programming

**Object navigator**

```html
document.writeln("navigator.platform = " + navigator.platform);
document.writeln("<br />");
document.writeln("navigator.plugins = " + navigator.plugins);
document.writeln("<br />");
document.writeln("navigator.userAgent = " + navigator.userAgent);
document.writeln("<br />");
</script>
</p>
</body>
</html>
```

```
navigator.appCodeName = Mozilla
navigator.appName = Netscape
navigator.appVersion = 5.0 (Windows, en-US)
navigator.language = en-US
navigator.userAgent = [object Navigator]
navigator.platform = Win32
navigator.plugins = [object PluginArray]
navigator.userAgent = Mozilla/2.0 (Windows, U; Windows NT 5.1; en-US; rv:1.8.1.7) Gecko/20070324 Firefox/2.0.0.7
```
Ajax Technology in Web Programming

More Information

- [http://www.alvit.de/handbook/](http://www.alvit.de/handbook/)
- [http://www.w3schools.com/](http://www.w3schools.com/)
- [http://www.w3.org/](http://www.w3.org/)