DOM

Sergio Luján Mora

Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

Contents

- What is DOM?
- DOM Levels
- DOM Level 0
- DOM Level 1

What is the DOM?

- The Document Object Model is an API for HTML and XML documents.
- It does two things for web developers:
  - it provides a structural representation of the document, 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.

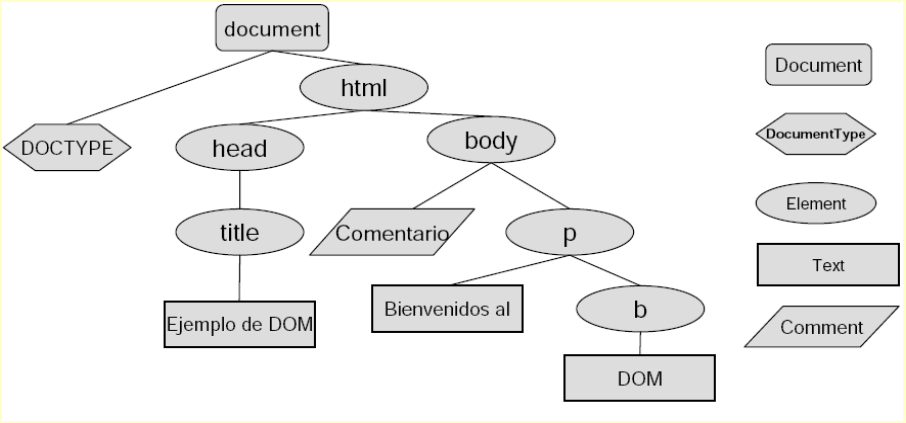
What is the DOM?

- Example:

```
<!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>
```

What is the DOM?

- Nodes reflect content and structure of the document:

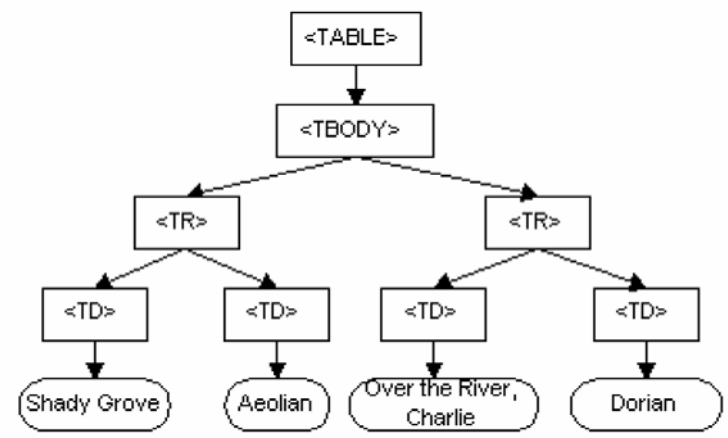


What is the DOM?

- Example:

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

What is the DOM?



What is DOM?

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

DOM Levels

- DOM 0
- DOM 1: Core and HTML
- DOM 2: Core, HTML, Style, Events, View, etc.
- DOM 3: in progress

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:

```
 // 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:

```
<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

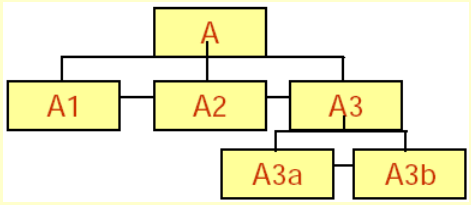
- Every node is a type of Node, but there are different subtypes: Document, DocumentType, Element, Text,Comment, …
- Attributes are nodes of type Attr, but they are not in the tree that represents the document

DOM Level 1

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

DOM Level 1

- Example:



A.firstChild = A1
A.lastChild =A3
A.childNodes.length = 3
A.childNodes[0] = A1
A.childNodes[1] = A2
A.lastChild.firstChild = A3a
A3b.parentNode.parentNode = A
A1.nextSibling = A2
A3.prevSibling = A2
A3.nextSibling = null

DOM Level 1

- getElementById(“elementID”)
  - `var element = document.getElementById(“myTable”)`
- getElementsByTagName(“tagName”):
  - `var images = document. getElementsByTagName(“img”)`
  - The special value "*" matches all tags.

DOM Level 1

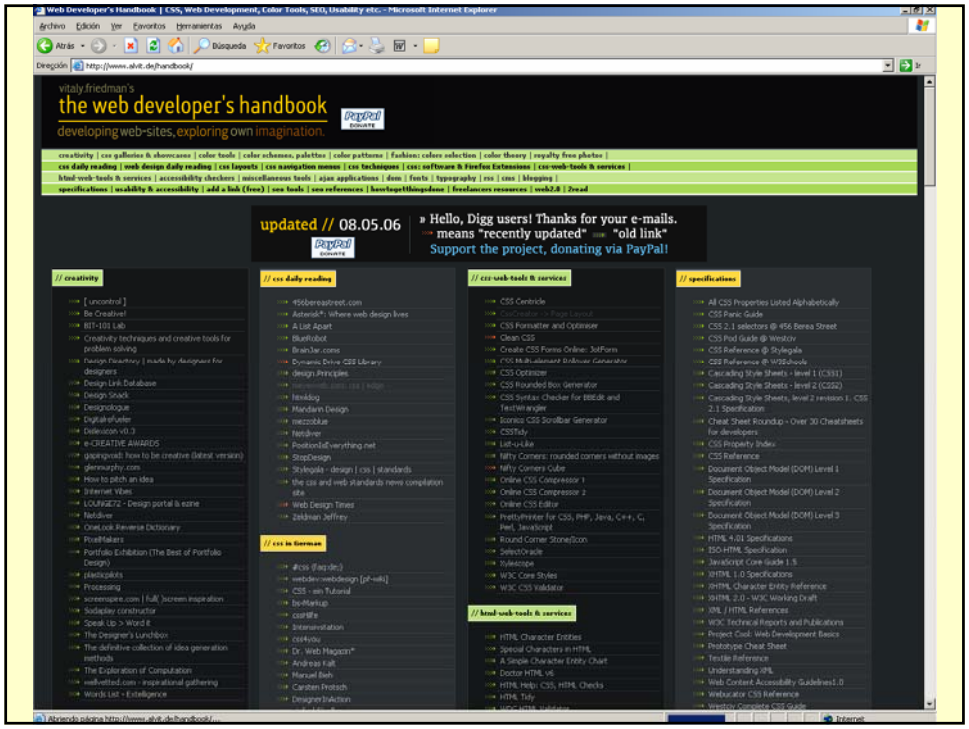
- createElement(“tagName”)
  - `var element = document.createElement(“tbody”)`
- appendChild(element)
  - `element.appendChild(newChild);`
- replaceChild(element, element)
  - `element.replaceChild(newElement, oldElement);`

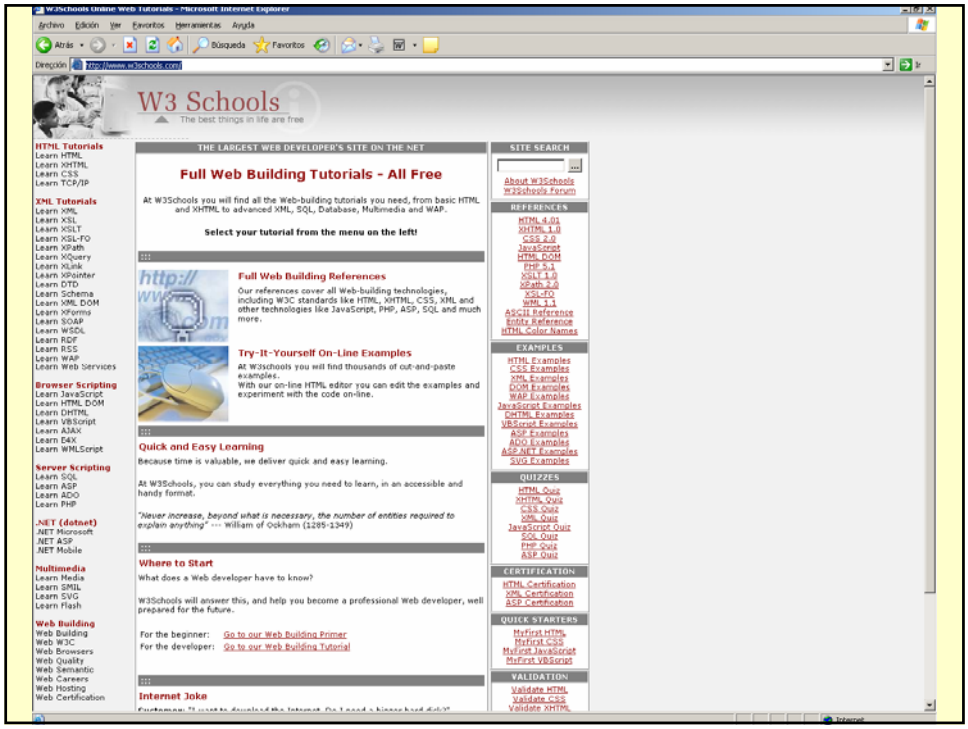
DOM Level 1

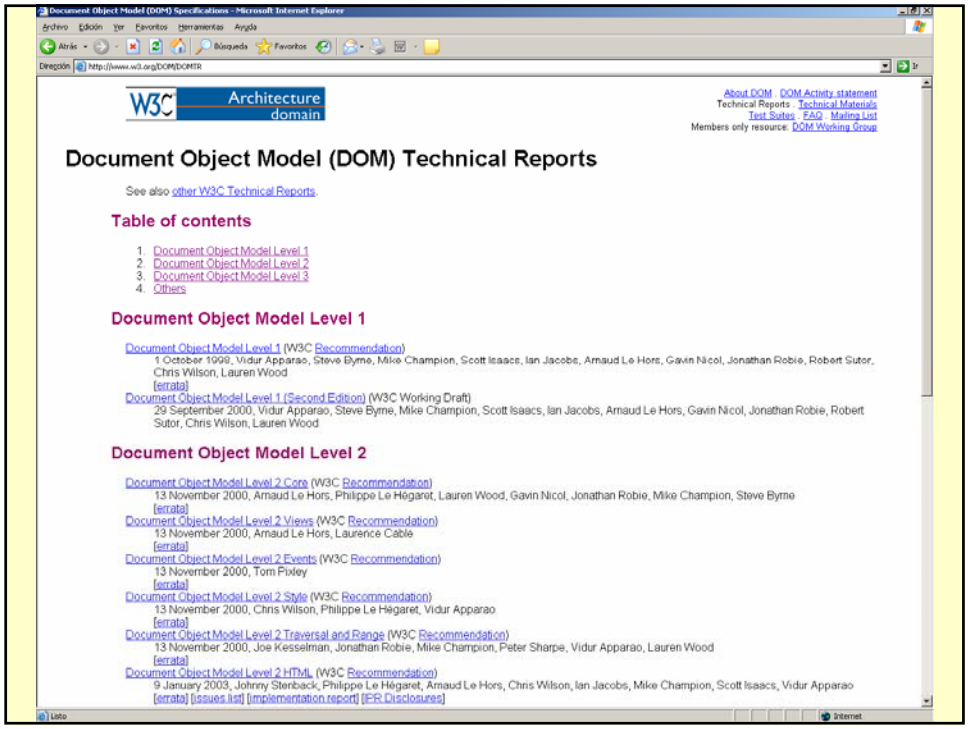
```
// HTML:
// <div id="d"><p>Content</p>
// <p>Further Elaborated</p>
// </div>

d = document.getElementById("d");
alert(d.innerHTML);
```

More Information

- http://www.alvit.de/handbook/
- http://www.w3schools.com/
- http://www.w3.org/




W3 Schools
The best things in life are free
THE LARGEST WEB DEVELOPER'S SITE ON THE NET
Full Web Building Tutorials - All Free
At W3Schools you will find all the Web-building tutorials you need, from basic HTML and XHTML to advanced XML, SQL, Database, Multimedia and WAP.
Select your tutorial from the menu on the left!
Full Web Building References
Our references cover all Web-building technologies, including W3C standards like HTML, XHTML, CSS, XML and other technologies like JavaScript, PHP, ASP, SQL and much more.
Try-It-Yourself On-Line Examples
At W3schools you will find thousands of cut-and-paste examples.
With our on-line HTML editor you can edit the examples and experiment with the code on-line.
Quick and Easy Learning
Because time is valuable, we deliver quick and easy learning.
At W3Schools, you can study everything you need to learn, in an accessible and handy format.
"Never increase, beyond what is necessary, the number of entities required to explain anything" --- William of Ockham (1285-1349)
Where to Start
What does a Web developer have to know?
W3Schools will answer this, and help you become a professional Web developer, well prepared for the future.
For the beginner: Go to our Web Building Primer
For the developer: Go to our Web Building Tutorial
Internet Joke
HTML Tutorials
XML Tutorials
Browser Scripting
Server Scripting
.NET (dotnet)
Multimedia
Web Building
SITE SEARCH
REFERENCES
EXAMPLES
QUIZZES
CERTIFICATION
QUICK STARTERS
VALIDATION


Architecture domain
About DOM . DOM Activity statement
Technical Reports . Technical Materials
Test Suites . FAQ . Mailing List
Members only resource: DOM Working Group
Document Object Model (DOM) Technical Reports
See also other W3C Technical Reports.
Table of contents
1. Document Object Model Level 1
2. Document Object Model Level 2
3. Document Object Model Level 3
4. Others
Document Object Model Level 1
Document Object Model Level 1 (W3C Recommendation)
1 October 1998, Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson, Lauren Wood
[errata]
Document Object Model Level 1 (Second Edition) (W3C Working Draft)
29 September 2000, Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson, Lauren Wood
Document Object Model Level 2
Document Object Model Level 2 Core (W3C Recommendation)
13 November 2000, Arnaud Le Hors, Philippe Le Hégaret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, Steve Byrne
[errata]
Document Object Model Level 2 Views (W3C Recommendation)
13 November 2000, Arnaud Le Hors, Laurence Cable
[errata]
Document Object Model Level 2 Events (W3C Recommendation)
13 November 2000, Tom Pixley
[errata]
Document Object Model Level 2 Style (W3C Recommendation)
13 November 2000, Chris Wilson, Philippe Le Hégaret, Vidur Apparao
[errata]
Document Object Model Level 2 Traversal and Range (W3C Recommendation)
13 November 2000, Joe Kesselman, Jonathan Robie, Mike Champion, Peter Sharpe, Vidur Apparao, Lauren Wood
[errata]
Document Object Model Level 2 HTML (W3C Recommendation)
9 January 2003, Johnny Stenback, Philippe Le Hégaret, Arnaud Le Hors, Chris Wilson, Ian Jacobs, Mike Champion, Scott Isaacs, Vidur Apparao
[errata] [issues list] [implementation report] [IPR Disclosures]

Exercises

- Sort a table:
  1. Get the table → getElementById
  2. Get the tbody → getElementsByTagName
  3. Get the rows → getElementsByTagName
  4. Insert the rows in an array (use cloneNode)
  5. Sort the array with sort() method of array
  6. Reverse order: use reverse() method of array
  7. Create a new tbody → document.createElement
  8. Append ordered rows → appendChild
  9. Replace old tbody → replaceChild

Exercises

- Add rows to a table

Exercises

```
<html>
<head>
<title>Add rows</title>
<script language="JavaScript">
function addRow()
{
  var table = document.getElementById('myTable');
  var row = table.insertRow(table.rows.length);
  var cell = row.insertCell(row.cells.length);
  cell.appendChild(document.createTextNode("Una prueba"));

}
</script>
</head>
<body>

<table id="myTable">
</table>

<form name="myForm">
<input type="button" value="Add row" onclick="addRow();">
</form>
</body>
</html>
```

Exercises

- Filter a table