Document Object Model
Agenda

• Understand the DOM model
  – How to walk it, search it, etc.
What is the DOM?

• Acronym for Document Object Model
• It is an API
  – And API for what?
• It is the browser API
  – A (mostly) common standard API across all browsers
  – It is the API for how to interact with:
    • What is rendered (drawn) to the window
      – And where
    • GUI events such as mouse movements and clicks
The DOM is a tree

Eg: [http://www.andrew.cmu.edu/course/67-328/examples/dom.html](http://www.andrew.cmu.edu/course/67-328/examples/dom.html)
The end.
Important DOM Globals

• window
  – Central browser global variable
    • Show in Chrome console

• navigator
  – Actually window.navigator, but unique ids are in global scope
  – Not that useful: browser information
    • Show in console

• screen
  – User's display properties

• document
  – I.e. window.document
  – The root of the HTML document
• If document IDs do not clash with any global variable, then those ID have global scope.
• E.g.
  
  \(<h1 id="hw">Hello Class</h1>\)

  "hw" is a global variable

• \(<h1 id="screen"> would not work (name clash)
• If more than one ID with the same name, an array is returned
  – E.g. "span1"
Walking the DOM

- Experiment with...
  - `document.body.childNodes[1]`
  - `document.body.firstChild`
  - `document.body.firstChild.nextSibling`
  - `document.body.firstElementChild`
  - `document.body.firstElementChild.innerHTML`
  - `document.body.firstElementChild.nodeName`
  - `document.body.firstElementChild.id`
  - `hw.nodeName`
  - `hw.nextElementSibling`
Challenge

Without using a search, find references to:

1. "within a div"
2. The text "bold"
3. The document title
4. The 2nd span element
5. Change the text "bold" to "windmill"

Eg: [http://www.andrew.cmu.edu/course/67-328/examples/dom.html](http://www.andrew.cmu.edu/course/67-328/examples/dom.html)
There are 3 ways to search the DOM:

• `document.getElementById("pp1").innerHTML`
  - "This is a paragraph with <b>bold</b> and <span id="span1">spanned<br>text within it""

• `document.getElementsByName("2nd")`
  - `[<span id="span2" name="2nd">spanned<br></span>]`
  - Notice the plural Elements and that an array is returned
    • (Even if only for one element)

• `document.getElementsByTagName("span")`
  - `[<span id="span1">Spanned<br>text</span>, <span id="span2" name="2nd">spanned<br></span>]`
Some useful nodeTypes

<table>
<thead>
<tr>
<th>Type</th>
<th>nodeType</th>
<th>nodeName</th>
<th>nodeValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>1</td>
<td>tagName</td>
<td>null</td>
</tr>
<tr>
<td>Attribute</td>
<td>2</td>
<td>Name of attribute</td>
<td>Value of attribute</td>
</tr>
<tr>
<td>Text</td>
<td>3</td>
<td>&quot;#text&quot;</td>
<td>Content of text node</td>
</tr>
<tr>
<td>CDATA Section</td>
<td>4</td>
<td>&quot;#cdata-section&quot;</td>
<td>Content of CDATA section</td>
</tr>
</tbody>
</table>

- If you have XHTML that you don’t know the structure of, you can test nodes and parse it.

- if (something.nodeType==3) {something.nodeValue="New text"}
Creating Elements & Attributes

• Create a new element:
  – `de=document.createElement("p");`
• Create a textNode
  – `tn=document.createTextNode("It is 11:30am.");`
• Add the textNode to the `<p>` element
  – `de.appendChild(tn);`
• Insert a node before another
  – `tn=document.createTextNode("What @me is it? ");`
  – `de.insertBefore(tn, de.firstChild);`
• Add the whole thing to the document
  – `var dv=document.getElementsByTagName("div");`
  – `document.body.insertBefore(de,dv[0]);`
• **Give it a try with the dom.html example file**
Practice

• Start with:
  – http://www.andrew.cmu.edu/course/67-328/examples/dom.html

• Change
  – “This is a paragraph with **bold** and spanned text within it”

• By making “paragraph” italic, resulting in
  – “This is a *paragraph* with **bold** and spanned text within it”
  – Constraint: Don’t alter “**bold** text within it”

• Hint
  – You need to add additional nodes.
var pp=document.getElementById("pp1");
pp.firstChild.nodeValue="This is a "
var it=document.createTextNode("with ");
pp.insertBefore(it,pp.firstChild.nextSibling);
var ie=document.createElement("i");
it=document.createTextNode("paragraph ");
ie.appendChild(it);
pp.insertBefore(ie,pp.firstChild.nextSibling);
Practice / Homework

• Start with example code from:
  – http://www.andrew.cmu.edu/course/67-328/examples/tablerow2.html
    • This is adapted from: http://www.adp-gmbh.ch/web/js/elements/createelement.html

• Modify the HTML and JavaScript code to:
  – Add a 3rd column (Column C)
    • Add a form element to input the value of Column C.
    • Add Column C to the table
    • Add code to add the value for Column C into the DOM
    • You don't have to add a button to add columns
  – Add a button to add a table border
    • Hint, to add an attribute (e.g. onClick) to an element, use:
      `element.setAttribute(attribute, value)`
  – 5 point bonus: Add a button to add additional columns

• DUE: Monday, September 14, before class

• Collaboration:
  – For this homework: it is ok to work in pairs
  – It is not OK to just copy from another
  – If working in a pair, only submit to Blackboard once, and indicate both names.
Definitive DOM source

• http://www.w3.org/DOM/
• But there are plenty of more readable tutorials, references, and examples.
  – Including JavaScript: The Definitive Guide
Homework for Monday

• You may work in pairs
• I'll post to the web site later today

1. Method chaining exercise
2. Tick tock exercise
3. Add column to a table exercise