Announcements

- Try to get help from me and tutors
- Today: DOM, JavaScript debugging
- Break around 10:15am
Document Object Model (DOM)

- With the HTML DOM, JavaScript can **access** and **change** all the **elements** in an HTML document.
- HTML documents are modeled in memory using a tree structure called the DOM:
  - Each document has a single parent node and this node can have children.
  - These nodes in turn can be parents of other children.
  - Each node (except the head node) has one, and only one parent.
  - All nodes in the tree can have 0 or more children.
  - DOM trees contain several kinds of node, the most common of which are element nodes (tags) and text nodes (content inside tags).
- See [DOM Intro](#) in the “JS HTML DOM” section of JavaScript Tutorial.

- **Example**
  - Open an HTML file (e.g., `js1.html`) on Chrome and view it in the “Elements” tab of Chrome’s developer tools (via “Inspect Element” on right click on the page).
DOM (cont.)

- DOM is a technology- and language-independent API for interacting with HTML documents

- The API allows developers to access nodes in the document, traverse the document, and manipulate elements in the document

- Each element in the DOM tree is represented by an object

- These objects support an API that allows them to be manipulated
DOM API in action

- Open the Chrome Console and type
  > document

- This will return the DOM representation of the HTML document, which is the exact same representation we saw from the “Elements” tab with “Inspect Element”

- The DOM API allows us to access elements, e.g., with js1.html
  > document.getElementById('aaa')

- In addition to querying, traversing, and manipulating the document, the DOM API also allows event listeners to be added to the nodes in the document
  - These can be used to detect users clicking buttons, or text being typed into input fields, along with many other types of events.
  - For examples, see “DOM EventListener” under JS HTML DOM in W3Schools' JavaScript Tutorial
JavaScript debugging

- Load `debugging/bmi.html` on Chrome (it will load `bmi.js` as well)
- Next, start the debugger
  - Command+Option+i on OSX
  - F12 or Cntl+Shift+i on Windows
  - Or, right-click and Inspect Element on either OSX or Windows
- Click on “Sources” tab; “Hit Cmd+P” (on OSX) and open `bmi.js`
  - Be sure that you are seeing a page with JavaScript code
- Set a break point by clicking on the line number on the left margin
- Run your app to the break point
- You can view/check the usual things that a debugger lets you
  - Call stack
  - Scope variables
  - You can try things on the “Console” window on a break point
  - etc.
JavaScript debugging (cont.)

• Step through code with the controls
  • Resume script execution
  • Step over next function call
  • Step into next function call
  • Step out of current function
  • Deactivate breakpoints
jQuery

- Is a lightweight, “write less, do more”, JavaScript library
- Is one of the most popular JavaScript frameworks
  - “Used by over 60% of the 10,000 most visited websites” [Wikipedia]
- You need to know the following to study jQuery
  - HTML
  - CSS
  - JavaScript
- We will not cover it in class and you won’t have to use it this semester, but keep in mind if you become serious about web programming (If time allows, we will come back to this later in the semester.)
- For more info on jQuery see
  - http://www.w3schools.com/jquery/default.asp