Lecture 3 – JavaScript & DOM

Web Application Development

January 20, 2015

Jeffrey L. Eppinger

Professor of the Practice

School of Computer Science

Lecture Schedule – 1ˢᵗ Half

(subject to change)

#1	Intro	#9	Django Templates
#2	HTML & CSS	#10	Images
#3	JavaScript & DOM	#11	AJAX
#4	Bootstrap	#12	jQuery
#5	HTTP & Django	#13	Databases
#6	Django Models	#14	Cloud Deployment
#7	Cookies & Sessions	#15	SSL
#8	Transactions	#16	Project Proposals

JavaScript

- JavaScript is like Java
  - "C" style syntax
  - Garbage collected
  - Object-based

- JavaScript is not Java
  - Functions are objects
  - Objects have named properties
  - Functions can be properties
  - Prototype-based
  - JavaScript is interpreted
  - JavaScript is dynamically typed
  - Runs in browsers (without special plug-ins)

Basic I/O

- `alert` puts up a text box, handy for debugging

  ```
  alert("Houston, we have a problem")
  ```

- `prompt` puts up a text box and waits for input

  ```
  prompt("'sup?")
  ```

- `console.log()` function appends to browsers console log

  ```
  console.log("Even handier for debugging")
  ```

- `document.write()` appends to the html doc!

  ```
  document.write("<h2>Haha</h2>")
  ```

- Note that you can use single or double quotes for strings

Where the JavaScript Goes

- Declare it in the document
  - Declare it in **`<script>`** **`</script>`** tags
  - Declare it in a separate .js file
  - Typically, put the **`<script>`** tags containing functions or inclusion of files containing functions in the **`<head>`** section of the HTML document
- Example
  - primes.html

JavaScript types

- string
  - Delimited with quotes or apostrophes

- number
  - Always floating point, but there are functions to parse and manipulate integers

- boolean
  - `true` and `false`

- function
  - Can be named: `function f(x) { return 2*x; }`
  - Can be unnamed: `var f = function(x) { return 2*x; }`

- object
  ```
  var pres = { first:"Barack", last:"Obama"}
  pres.age = 53;
  ```
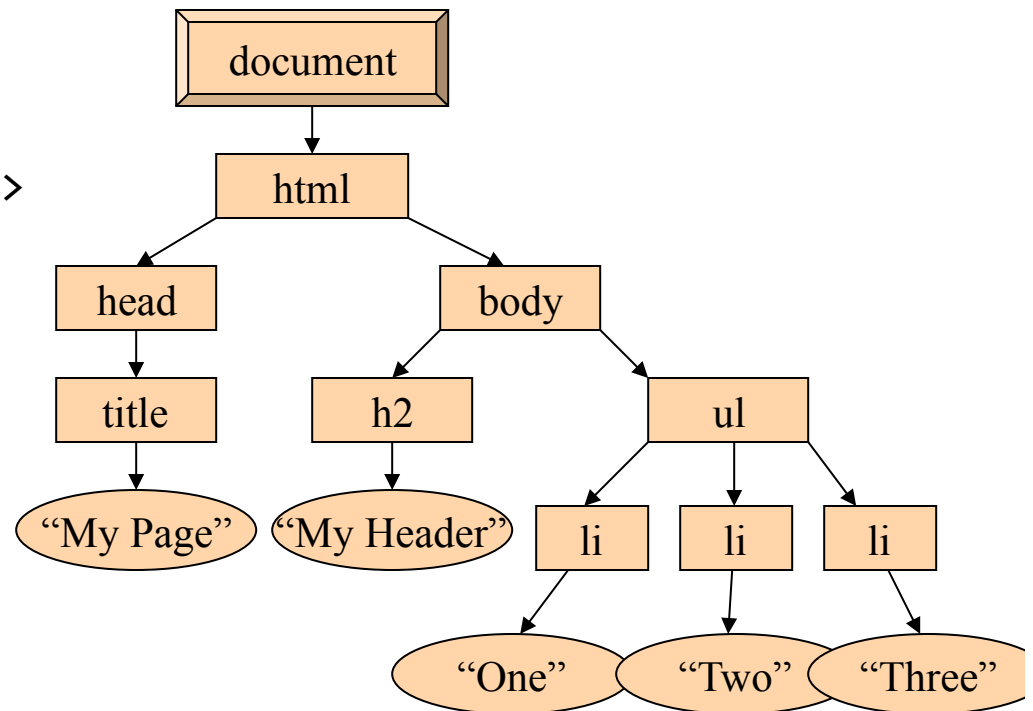
JavaScript Events

- You can add declare events on HTML elements

  ```
  onload
  onclick
  onfocus / onblur
  onmouseover
  ```

- Provide a JavaScript method which will be invoked when the event occurs

- Example: onload in index.html

  ```
  <body onload="document.getElementById('last').focus()">
  ```

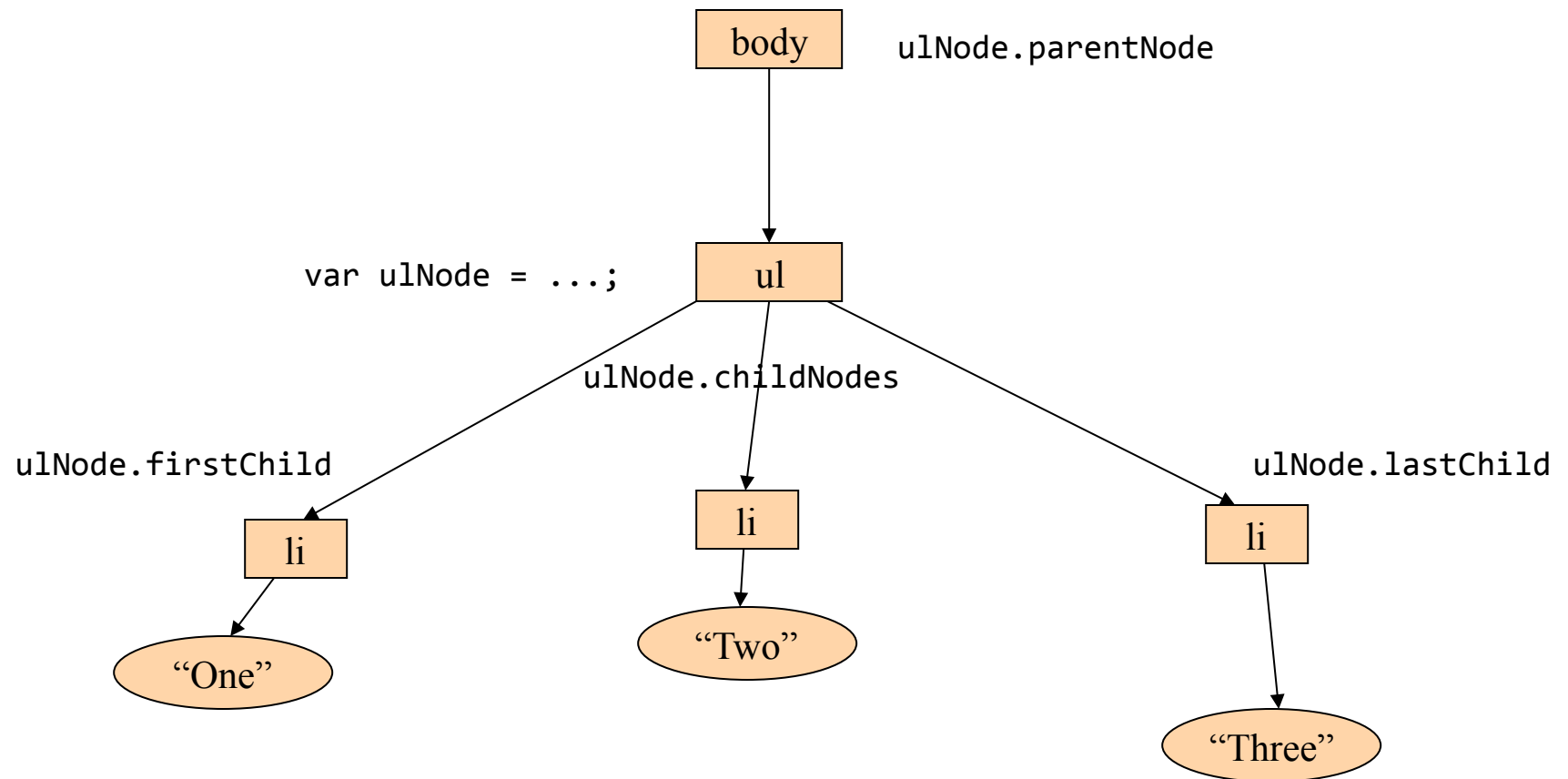# Document Object Model

- A tree of nodes that represent the HTML document
- Example:

```
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <h2>My Header</h2>
    <ul>
      <li>One</li>
      <li>Two</li>
      <li>Three</li>
    </ul>
  </body>
</html>
```

Looking around the DOM Tree

Copyright (c) 2015 J.L. Eppinger

DOM Elements

- ## Nodes in the DOM tree
  - Three types: element, attribute, text
- ## Use fields to obtain parents, children, siblings
  - `parentNode, firstChild, lastChild, childNodes, nextSibling, previousSibling`
- ## You can apply methods to change the element
  - Insert or delete children
    - `appendChild()`, `deleteChild()`
  - Modify the elements attributes
    - `onclick, href, style, className, id, name`
  - This changes the HTML document!

The Document Object

- Is a predefined JavaScript variable (`document`)
- It's the root of the DOM tree
- Has methods that allow you to search the tree and obtain node objects that represent elements in the HTML
  - `getElementById()`, `getElementsByName()`, `getElementsByTagName()`
  - Example: `var ulNode = document.getElementsByTagName("ul")[0];`
- Has methods that allow you to create additional elements
  - `createElement()`, `createTextNode()`
- You can inspect the document itself
  - `url`, `cookies`
- You can modify attributes of the document
  - `bgColor`, `cookies`

Examples

- index.html
- movement.html
- todolist.html & todolist.js

Beware of "specialness" of JavaScript

- Symantics of ==
  - So there is a ===
- Objects have constructors and prototypes
  - It has a full-blown mechanism, but it's very different than Java's
- It's interpreted
  - So you'll see errors at runtime
- Complex variable scope rules

Debugging

- Firefox
  - Tools => Web Developer

- Chrome
  - View => Developer

Alternatives

- Adobe Flash
  - Browser plug-in

- Google Web Toolkit (GWT)
  - Write your client-side app using Java in Eclipse
  - GWT will compile it into JavaScript & HTML

- Google Dart Programming Language
  - Compiles to JavaScript
  - They say it will some day run native in Chrome

Web Resources

- [w3schools Javascript tutorial](#)
  - Google JavaScript topic => w3schools page
- [MDN (Mozilla Developer Network) JavaScript Guide](#)
- [MDN (Mozilla Developer Network) JavaScript Reference](#)
- [MDN Introduction to Object Oriented Javascript](#)

Homework #2

- To be posted tonight
- Implement your HW#1 calculator using JavaScript
- Due on Monday, January 26th