Intro to Web Programming
Week 2

January 14, 2014

JavaScript (JS)

- A “programming language commonly used to create interactive effects within web browsers”
- Client-side, interpreted, dynamically typed
- C-like syntax
- Has nothing to do with Java, even if its name suggests that it should

Some things you can do with JS

- “anything from simple things to implementing GoogleDocs”
- Suggest the complete term a user might be entering in a search box as she types
- Show and hide content based on a user clicking a link or heading, to create a “collapsible” content area

DOM

- Document Object Model
- Standardized list of web page elements that can be accessed and manipulated using JavaScript

jQuery

- A JavaScript library intended to make JavaScript programming easier
- Also helps solve some cross-browser problems

Embedding JS into HTML

- Use the <script> tag
- Often done in the <head> part of the html file, but can be inserted anywhere
- Can be embedded directly or written into a separate file

  ```
  <script src=“navigation.js”></script>
  ```
- Can include an arbitrary number of JS files; can have a combination of external files and embedded JS.
  - Remember that each request of an external script slows down performance

Quick Overview of JS Basics

- Case-sensitive
- Tabs and whitespace ignored (unless part of a string)
- Statements terminated with semicolons
  - Line breaks can also work, but try to follow the semicolon standard
- Comments as in Java
  - Single line //
  - Multi line /*   */

JS Basics, cont'd: Variables

- var itemsOrdered;
- var itemsOrdered = 5;
- Variable types
  - number (can perform the usual arithmetic operations on these; can also use +=, ++, --)
  - string (put string literals in quotes; same concatenation rules as in Java)
  - booleans (true, false. null, undefined, 0, and empty strings are inherently false; every other value is inherently true)

JS Basics, cont'd: Variables

- Arrays
  - Square brackets
  - 0-indexed
  - Values need not be all of one type
  - Think of them as Python lists
  - `var items = [“cakes”, “hello”, 5.3, 9];`
  - `items[2]` has the value 5.3
  - `items.length` has the value 4

JS Basics, cont'd: Comparison Operators

- == equal to
- != not equal to
- === identical to (equal and of the same type)
- !== not identical to
- >, >=, <, <=
- Note that `“5” == 5` is true, but `“5” === 5` is false

JS Basics, cont'd: if-else statements

- Use basic Java syntax:

```
if ( cond ) {


} else {


}
```

JS Basics, cont’d: loops

- for
- while
- Again, follow familiar Java syntax

DOM

- Document Object Model
- Gives us a way to access and manipulate the contents of a document.
- An API for HTML and XML pages.
- Everything from the doctype to each individual letter can be accessed via the DOM and manipulated with JavaScript.

DOM

- One way to think of the DOM is in terms of the document tree.
  - Each element on the page is a node in the tree.
- The document object identifies the page itself.
- Useful properties and methods.
  - E.g., `document.getElementByID(“novice”).innerHTML`
  - finds the element with the id “novice” and then gets the HTML content within that element.

Sample DOM methods

- getElementsbyTagName(“p”) returns every paragraph element, wrapped in a nodeList
- getElementsByClassName(“className”)
- getAttribute()
  - E.g.,

    ```
    var bigImage = document.getElementByID(“lead-
    image”);
    alert( bigImage.getAttribute(“src”) );
    ```

- bigImage.setAttribute(“src”, “flower.jpg”);

Setting Properties

```
var introDiv =
  document.getElementById(“intro”);

introDiv.innerHTML = “<p>This is new
  intro text.</p>”;

document.getElementById(“intro”).styl
  e.color = “#f58220”;
```