

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

## week9

Tommy MacWilliam

`tmacwilliam@cs50.net`

November 7, 2011

# Announcements

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ pset6
- ▶ CS50 seminars:

`https://manual.cs50.net/Seminars`

- ▶ Git Magic: Versioning Files Like a Boss. Sunday, November 13 at 3pm.
  - ▶ git checkout this seminar
- ▶ ^ I am aware of how shameless this is.

# Today

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ common pset6 mistakes
- ▶ JavaScript `<3333333333333333`
- ▶ DOM
- ▶ AJAX

# pset6

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ remember, `strlen` iterates over /every/ character in the input string
  - ▶ with a pointer to an array, no other magic way to determine its length
- ▶ this is slow!
  - ▶ particularly if it's in the condition of a for/while loop
- ▶ solution: look for the null byte (`'\0'`) as your condition instead of using `strlen`
  - ▶ only iterate over every character in the string once

# pset6

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ arrays just for the purpose of storing true/false values for elements are extremely wasteful
  - ▶ let's say you use an integer array where 1 represents true and 0 represents false
  - ▶ how many bits does it take to represent true or false? 1
  - ▶ how many bits are in an integer? 32
- ▶ we can do much better

# pset6

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ solution: bit vector
  - ▶ integer (or integer array) in which each bit represents a true/false value for an element
- ▶ bitwise operators ( $\&$ ,  $|$ , etc.) allow you to get and set bits in an integer
- ▶ example: `bitvector = 2529692;`
  - ▶ what elements are set?
  - ▶ `bitvector = 001001101001100110011100;`
  - ▶ set elements: 2, 3, 4, 7, 8, etc.
    - ▶ in this example, the 0th element is the rightmost bit
- ▶ have more than 32 elements? use an integer array
  - ▶ remember, arrays are adjacent in memory, so the 33rd bit is the first bit of the 2nd element

# pset6

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ so how much space are we really saving?
- ▶ let's say we have a hashtable with 10,000 elements and we have a perfect hash function
  - ▶ storing a `char*` at each cell requires  $4 * 10,000 = 40,000$  bytes
  - ▶ using a bitvector requires  $10,000 / 32 = 313$  bytes
- ▶ huge savings!

## ▶ other cool tricks

- ▶ GCC compiler optimizations: taking out the `-ggdb` flag and adding the `-ofast` flag allows the compiler to make your code faster for you
- ▶ `C register` keyword decreases the time it takes for the CPU to access a variable
  - ▶ example: `register int i = 0;`
- ▶ functions like `fgets` and `fgetc` are SLOW
  - ▶ use a single call to `fread` (or `mmap`) to read the entire dictionary into memory in one shot



# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript is the best programming language ever
  - ▶ other people will try to tell you otherwise
  - ▶ they are wrong

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript is NOT Java
  - ▶ Java is not the best programming language ever
- ▶ marketing ploy by Sun and Oracle
  - ▶ the “hot new web-programming language”

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ PHP: server-side
  - ▶ runs on server, produces output, browser downloads
- ▶ JavaScript: client-side
  - ▶ browser downloads, runs code

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ syntax (also) very similar to C and PHP
  - ▶ if, else, for, while, etc.
  - ▶ strings are built in (just like PHP)
  - ▶ variables don't need dollar signs (yay!)
- ▶ no types for variables or functions
  - ▶ `x = 5;`
  - ▶ `function increment(x) { return ++x; }`

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript can be inserted into your page using the `<script>` tag
  - ▶ `<script>` inside `<head>`: will be evaluated before page loads (used for functions and events)
  - ▶ `<script>` inside `<body>`: will be evaluated as page loads
- ▶ just like CSS, JS can also be placed in an external file using the `src` attribute (not `href`)
  - ▶ `<script src="script.js"></script>`
  - ▶ **CANNOT** say `<script src="script.js" />`
    - ▶ :(

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ example time!
  - ▶ `simple-javascript.html`

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ like PHP, JS arrays also do not have a fixed size and can contain different types
  - ▶ declaring an array: `var numbers = [ 1, 2, 3, "4", 5.3 ];`
  - ▶ adding a new element to an array:  
`numbers.push(6);`
  - ▶ removing and returning the first element of the array:  
`numbers.shift();`
  - ▶ removing and returning the last element of the array:  
`numbers.pop();`
- ▶ JS arrays are objects! (`push`, `shift`, and `pop` are functions in the `Array` class)

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ unlike PHP, JavaScript makes a difference between associative arrays (key/value pairs) and arrays
  - ▶ like PHP, different types can be present in an associative array
  - ▶ 

```
var person = { "name": "tommy", "rank": 1 };
```
  - ▶ 

```
person["occupation"] = "TF";
```



# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript associative arrays are also objects
  - ▶ `var object = { "name": "tommy", "rank": 1 };`
  - ▶ `object.occupation = "TF";`
- ▶ syntax is interchangeable!

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ being able to create flexible objects on the fly makes JS /extremely/ powerful
- ▶ 

```
var tf = { "name": "tommy", "grade":  
  function() {  
    alert("done!");  }  
};
```

  - ▶ the `tf` object has a name and a grade function
  - ▶ 

```
var name = tf.name; tf.grade();
```

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript provides the `for-in` loop for iterating over both arrays and associative arrays

```
for (var i in tf)
    alert(i + "is: " + tf);
```

- ▶ for an array, `i` will be an integer ranging from 0 to `array.length - 1`
- ▶ for an associative array, `i` will be the keys in the associative array
  - ▶ a hash CANNOT be accessed using numerical indexes (i.e. `tf[0]` is meaningless)

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ JavaScript scope is a bit different than other languages
- ▶ the `var` keyword limits a variable's scope
  - ▶ with `var`: scope limited to current function (not loop, as in C!)
  - ▶ without `var`: global scope

# JavaScript

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ example time!
  - ▶ `scope.html`

# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ the **D**ocument **O**bject **M**odel provides a way to access and manipulate HTML as objects
  - ▶ DOM is language independent
  - ▶ languages like JavaScript, Python, and Ruby have DOM implementations
- ▶ each tag is an object
  - ▶ attributes are properties of the object
  - ▶ contained tags are children of a parent tag

# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ in JavaScript, the DOM is loaded into the global `document` object
- ▶ `document.getElementById(id)`: get the element with the given ID
- ▶ `document.getElementsByTagName(tag)`: get an array of all elements with the given tag
- ▶ `document.anchors[]`: array containing all anchor tags on the page
- ▶ `document.images[]`: array containing all images on the page

## ► properties of JavaScript DOM objects

- `innerHTML`: text contained within the element
- `nodeName`: name of the tag of the element
- `parentNode`: parent of current element, represented as a DOM object
- `children`: array of child elements, represented as DOM objects
- `style`: object representing CSS properties of element
- `<attribute>`: each tag attribute has its own property



# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ example time!
  - ▶ `dom.html`

# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ the DOM also provides us a way to attach events to elements
  - ▶ event is a user interaction, like a mouse click
- ▶ common events
  - ▶ `onclick`: mouse click
  - ▶ `ondblclick`: double click
  - ▶ `onmouseover`: mouse moves over an object
  - ▶ `onmouseout`: mouse moves off an object
  - ▶ `onkeypress`: user pressed a key on the keyboard
  - ▶ `onload`: DOM has finished loading
  - ▶ `onchange`: value of a form changes
  - ▶ `onsubmit`: form is submitted

# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ function that fires when event occurs is called an “event handler”
- ▶ two ways to attach event handlers to elements
  - ▶ JavaScript: get DOM object, then add property for event name
  - ▶ HTML: use attributes like `onclick` and `onmouseover` and set value equal to name of function

# DOM

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ example time!
  - ▶ `events.html`

# AJAX

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ **Asynchronous JavaScript And XML**
- ▶ allows a page to make dynamic HTTP requests without reloading the page
- ▶ popularized in the early 2000s with Microsoft Outlook Web Access and Gmail
- ▶ very widely employed by “Web 2.0” and modern web apps

# AJAX

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ **AJAX functionality provided by the `XMLHttpRequest` class**
- ▶ **making an AJAX request**
  - ▶ create an `XMLHttpRequest` object
  - ▶ construct the URL to make the request to
  - ▶ create an event handler to handle the server response
  - ▶ open/send the request

# AJAX

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ an AJAX request has 5 states
  - ▶ 0: not initialized
  - ▶ 1: connection established
  - ▶ 2: request received
  - ▶ 3: answer in progress
  - ▶ 4: done

# AJAX

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ HTTP status codes tell browser the result of the request
  - ▶ 200: OK
  - ▶ 301: Moved permanently
  - ▶ 401: Unauthorized
  - ▶ 403: Forbidden
  - ▶ 404: Not found



# AJAX

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

▶ example time!

▶ `ajax.html, time.php`

# Practice Problems

week9

Tommy  
MacWilliam

pset6

JavaScript

DOM

AJAX

Practice  
Problems

- ▶ make a link grow/shrink when the user hovers over it
- ▶ implement the Twitter textbox character counter
- ▶ change page to a random color on click