Week 2 Lecture 4

Applied

What’s in this lecture?

- Data Structures in JavaScript:
  Arrays and Hashes

What’s a Data Structure?

- A Data Structure is a *thingy* that holds data
- Depending on the shape of the data structure, it has different speeds for different operations
- Typical operations are: insert(x), remove(x), contains(x)

Data Structure Analogy

- Things that hold stuff: Bookshelf, Paper Bag, Handbag, Backpack, Cubby, Dresser, Pile of Laundry
- A library has bookshelves ordered by Dewey Decimal or Library of Congress System: how are find(book), insert(book), remove(book) implemented?
- A Dry Cleaner has an enormous circular rack ordered by (last name, first name): how are find (clothes), insert(clothes), remove(clothes) implemented?

Array

- An array is a data structure that holds a fixed number of elements (similar to a linear row of cubby holes)
- Accessing any element of an array by its position number (or *index*) is very fast
- Assigning and retrieving elements from an array in JavaScript looks like:

  ```
  a[0] = 3;        a[1] = 4;       a[2] = 2;
  var x = a[2];
  ```

Random-Access Memory

- Memory in modern computers is set up as a huge array of memory cells that each hold 1 byte (8 bits) of data
- Each cell has an *address*, from 0 to 4 billion (if there is 4-GigaBytes RAM) or more (or less for older machines)
- Accessing any memory cell by its address is very fast
- Each array object has a *base address*
- To find element N of an array, its address is base_address + (N * element_size)

Arrays in JavaScript

```
var v = Array.new;             // v = []
v.push(3); v.push(2);          // v = [3, 2]
var x = v.pop();               // x = 2; v = [3]
var y = v.length;              // y = 1
var z = v[0];                  // z = 3; v = [3]
```

Hash

- A hash (or associative array) is an object that makes it easy to associate a *key* with a *value*
- Think of the dry cleaner: clothes are sorted by customer, not by type
- Assigning and retrieving elements of a hash looks like:

  ```
  clothes[“joe”] = “sweater”;
  clothes[“bob”] = “suit jacket”;
  var z = clothes[“joe”];  // z = “sweater”
  ```

Hashes in JavaScript

```
var tabby = new Object();
c[“name”] = “tabby”;
c[“type”] = “cat”;
c[“says”] = “meow”;

var rover = new Object();
c[“name”] = “rover”;
c[“type”] = “dog”;
c[“says”] = “woof”;
```

Using Hashes

```
function animal_speak(p) {
  alert(p[“name”] + “ says “ + p[“says”]);
}

animal_speak(tabby);  // ?
animal_speak(rover);  // ?
```

Iteration over Arrays

```
// *a* is an array; going forwards...
for (var i = 0; i < a.length; i++) {
  var current = a[i];
  // do stuff with current ...
}

// backwards...
for (var i = a.length - 1; i >= 0; i--) {
  var current = a[i];
  // do stuff with current ...
}
```

Iteration over Hashes

```
// *a* is a hash; *i* gets value of each *key*
for (var i in a) {
  var current = a[i];    // current is *value*
  // do stuff with current ...
}
```

Array and Hash Literals

```
var a = [1, 2, 3];
var a = new Array();
a.push(1); a.push(2); a.push(3);

var h = { “joe” : “sweater”, “bob” : “t-shirt”};
var h = new Object();
h[“joe”] = “sweater”; h[“bob”] = “t-shirt”;
```

Exercises

- Write a JS function that takes an integer argument N, and creates an array containing the integers from 1 to N
- Write a JS function that takes form values and creates a hash containing "person" attributes, such as name, age, and location
- Write a JS function that takes the "person" hash and displays it in a DIV element on the current HTML page