Javascript Programming I

The Basics
Goals

• Introduce Javascript
• Review Basics of Javascript Language
Javascript is ...

- NOT Java
- THE MOST popular language on GitHub... by far
Javascript is...

• a way to add interactive content to web pages
• a scripting language with an interpreter (does not compile to an intermediate language!)
• the language of the browser
• sometimes painful to program in
• I’m going to assume a basic working knowledge of
  – HTML
  – CSS
• If you need a refresher...

http://www.w3schools.com/
Web Architecture

Client (browser)

Server

Server Side Code (php, servlets, etc.)

Database

- url
- server requests
- query
- data

html + JS

- html + JS
• JS enables your browser, the client, to interact with the user without making a “round-trip” to the server
• JS gives you, the developer, access to the HTML Document Object Model (DOM) which structures the page
  – Catch and respond to user interface events
  – Insert elements into the page
  – Change or remove elements
<!DOCTYPE html>
<html>
<head>
  <meta charset=utf-8>
  <title> JS HelloWorld! </title>
  <script src="helloWorld.js"></script>
</head>
<body>
  Body of the HTML Document Goes here!
</body>
</html>
• JavaScript uses a C-like syntax, which requires the use of semicolons to delimit statements.
• JavaScript attempts to do auto-insertion and make them optional (argh!!!)
  – Put them in yourself after each statement!
• In JavaScript, a linefeed can be whitespace, or it can act as a semicolon. This replaces one ambiguity with another.

```javascript
return {
  status: true
};
```

```javascript
return {
  status: true
};
```
Types

- Single **Number** type
- **Strings**
  - No Characters
- **booleans, null, undefined**
- Numbers and Strings have methods

```javascript
alert("2.5.toPrecision(10) = " + 2.5.toPrecision(10));
alert("'myString'.toUpperCase() = " + "myString".toUpperCase());
```
Equality

= (assignment)
== (equality)
=== (strict equality)
Variables and Scope

• When defined inside a function a `var` is private

• if you forget the `var`, your variable is global, no matter where you define it

• blocks `{}` do not create new scope
  
  – define all variables at the top of functions

• More on Scope and Variables to come later!
Control

- if, then, else
- switch, case
- loops
  - while
  - do while
  - for
- try, return, break
Objects and Arrays

• Object Literals
  - Convenience for creating new objects
  - Defined as associative arrays (maps, dictionaries)
  - key, value pairs
  - ALWAYS passed by reference, never copied

```javascript
var olit = { "name": "ron", "age": 25};
alert("olit name = " + olit["name"]); //or olit.name
```
Array Literals

- Convenience for specifying new arrays
- Arrays are objects

```javascript
var alit = [1, 2, 3, 10];
```

- Can contain a mixture of types
- Can `delete` elements, however this leaves gaps
- Use `splice` to delete without gaps
- Rule of thumb: When property names are small sequential integers, use Array, else Object
Functions...

- are objects
  - therefore, a collection of name-value pairs
  - linked to prototype: `Function.prototype`

- Contain two additional properties
  - context
  - code to implement the behavior
```
var add = function foo (a, b) {
  return a + b;
};
```