CPSC-112 Introduction to Programming

JavaScript Introduction

Wednesday, April 16th 2014
Announcements

• Proposal progress
• Part 2 of PS8 will be released today
• Questions/Comments on GAE and Android
HTML, CSS Review Example

- Review HTML and CSS by building a basic HTML page in several steps.
JavaScript: Server-supplied code for Client-Side Execution
JavaScript: Big Picture

- **Server perspective:**
  - We can link to code files, just like CSS
  - “Alive” interfaces are cooler than static

- **Client perspective:**
  - I have a powerful CPU
  - Pictures and text are cool, but interfaces should react
  - Might as well give me some code to execute in addition to rendering HTML
JS: Link to it from within HTML

```html
<html>
  <head>
    <title>This is a title</title>
    <script src="jscode.js"></script>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```
JS: Fetched like any resource

GET jscode.js

HTTP/1.1 200 OK
JS: Fetched like any resource

index.shtml:
<html>
<head>
<title>CP</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div>
    <script src="jscode.js">
      F = function {
        ...
      }
    </script>
  </div>
</body>
</html>
Let's dive right in:

```javascript
function myFunction() {
  document.getElementById("demo").innerHTML = "My custom text!";
}
```

- Looks like Java object
- Method call on object
- String literal, Right-hand operand
- Local variable of the object to the left
- String literal
Let's dive right in:

```javascript
function myFunction() {
    document.getElementById("demo").innerHTML = "JavaScript is resetting this text!";
}
```

http://www.w3schools.com/js/tryit.asp?filename=tryjsExternalexample
JavaScript Syntax Overview


Linked on Course Schedule
JS: Values

<table>
<thead>
<tr>
<th>Type</th>
<th>Example of typed values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>42, 3.14</td>
</tr>
<tr>
<td>Logical (Boolean)</td>
<td>true / false</td>
</tr>
<tr>
<td>Strings</td>
<td>“Hello”</td>
</tr>
<tr>
<td>null</td>
<td>A special keyword, case-sensitive</td>
</tr>
<tr>
<td>undefined</td>
<td>Top-level property, primitive value</td>
</tr>
</tbody>
</table>

- All values can be stored in variables using 'var' keyword:

  ```javascript
  var my_variable = 32;
  ```

- In addition, **objects** and **functions** are fundamental components of the language
JS: Conversion

- Convert as you'd expect:
  ```javascript
  var x = 9;
  var y = "My number is: ";
  var z = y + x; // "My number is: 9"
  ```

- Variables are variables – of any type:
  ```javascript
  var x = 9;
  x = "and now I'm a String";
  ```

- Convert Strings to Numbers:
  ```javascript
  var x = parseInt("2") + parseFloat("3.3");
  ```
JS: Variables

- If no initial value, 'undefined' is default

```javascript
var input;
if(input == undefined){
    doThis();
} else {
    doThat();
}
```

- Global vars outside functions, local within

See JS comparison operators: http://www.w3schools.com/js/js_comparisons.asp
JS: Variable Scope

JS: Value Literals

- We've already seen integers, floats, Strings and even Booleans (true/false)
- String literals can use ' or "
- Array literals work in JS as well:
  ```javascript
  var coffees = [
    "French Roast", "Colombian", "Kona"
  ];
  ```
- Unspecified elements are 'undefined':
  ```javascript
  var fish = ["Lion", , "Angel"];"
JS: Object Literals

- The backbone of most JS code
- Simple, yet extremely powerful
- Basically just dictionaries
JS: Object Literals

- The backbone of most JS code
- Simple, yet extremely powerful
- Basically just dictionaries
JavaScript Examples
JS: Dev Console in Browsers

- Chrome (and firefox?) come with a powerful JavaScript interpreter:
  - One command per line (no multi-lines)
  - Full syntax support
  - Create vars, functions, objects
  - Alerts
JS: Browser “Loads” It

- Like a Java “import” statement
- The code is ready to be called from HTML
  - Bind the JS to mouse click events
JS: Some Notes

- A single page can load multiple .js files
- JS can be embedded inline (like CSS)
- JS executes within a browser “sandbox”
  - JS only accesses the current page (generally)
  - Contrast this with Java
- We'll learn about the global objects: `window` and `document` in the next lecture
JS: Light Bulb Example

- http://www.w3schools.com/js/tryit.asp?filename=tryjs_lightbulb

- Notice how the JS code is embedded directly in a `<script>` tag, rather than linked out to another file.
JS: Final Example

- Load the local example.html file and discuss the various ways to call JS methods.