JavaScript Week 2

TDD with Jasmine and Introduction to the DOM

From Last Week

- Go over Shopping Cart class from homework
- Go over prototype
- JavaScript version 1.6
- Global functions for converting strings to numbers: parseInt, parseFloat
    - http://www.w3schools.com/jsref/jsref_obj_global.asp

Test Driven Development (TDD)

- What is TDD?
- Write tests first to specify the behavior of the code
- Code until your tests pass

Jasmine

- Test framework for JavaScript with syntax similar to rspec for Ruby
- Written by and used by the guys at Pivotal

Activity

- Test our MagicalItem class

Scope

- No block level scoping
    - Since each function is an object, there is only 1 scope per function
    - So this is valid (but bad form):

```
function myFunction() {

    if (true) {
        var a = ‘a’;
    } else {
        var a = ‘b’;
    }

    console.log(‘a is: ‘, a);
}
```

The Document Object Model (DOM)

- Objects and object properties that make up an HTML document
- API for interacting with objects in an HTML document
- Each browser has a slightly different DOM!
- Platform and language neutral

Basic DOM Objects

- window - top level object, global scope
  - document - contains the elements that make up your document
  - location - information about the current page location
  - navigator - information about the web browser
  - frames - collection of frames and iframes
  - ...

Basic DOM Objects

- DIV - basic container
- Text Node - container for text, all text is contained in a text node
    - For example, all text contained inside a <p> tag is implicitly contained in a text node

Create and Access DOM Objects

- Objects assigned unique ID’s
    - document.getElementById(‘id’)
- Objects in a collection
    - Add to ‘childNodes’ collection using appendChild
    - You can then iterate over the ‘childNodes’ collection

Events and Listeners

- Each type of Object has an associated set of events that it “throws”
- You can assign any function to execute when an event is thrown
- To attach an event handler declaratively:

```
<body onload=”someFunction();”>
```

Event Handlers

- To attach an event handler declaratively:

```
<body onload=”someFunction();”>
```

- Attach using an inline function:

```
var button = document.createElement("button");
button.innerHTML = "Click Me";
button.onclick = function() {
                     alert('you clicked me!')
                 };
```

- Attach using a named function

```
function handleButtonClick() {
    alert('you clicked me!');
}

button.onclick = handleButtonClick;
```

the onload event

- You must wait for the window onload event before you can interact with the DOM.
- If you don’t wait for this you will get non-deterministic behavior.

Activity

- Build catalog of Items
  - Hard code list of MagicalItem objects
  - Onload iterate over list and create DOM nodes for each item

Homework

- Write Jasmine tests for ShoppingCart class
- Encapsulate the MagicalItem view nodes in a MagicalItemView class that knows how to create it’s own DOM nodes
- Add a button to each item that when clicked adds the item to the shopping cart
- Read chapter 2 until XHR (server requests)
- Read chapter 6 - execution speed