# JavaScript Week 2

## TDD with Jasmine and Introduction to the DOM

# 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);
}
```

# From Last Week

- Go over Shopping Cart class from homework

# 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
- http://github.com/pivotal/jasmine

# Activity

- As a group test our MagicalItem class
- Pair off and test the ShoppingCart class
- How does ‘beforeEach’ work if it is before a nested describe?

# The DOM

# 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 such as div, button, etc...
  - 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 DOM Objects

- Declaratively:
  - 
- Programmatically
  - document.createElement('div');

# 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 Handlers

- 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();">
```

# 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

- Create an HTML page that has one  element declared with id="container"
- Programmatically add one of each to this container div:
  - div
  - text
  - button

# Styling DOM nodes

- Style with CSS selector:

```
#someId {
    background-color:#ECECEC;
    font-size:14pt;
}
```

- Or style programmatically:

```
var myDiv = document.getElementById('myDiv');
myDiv.style.backgroundColor = "#ECECEC";
myDiv.style.fontSize = "14pt";
```

# Styling with CSS Classes

- You can define a CSS class like so:

```
.someClassName {
    background-color:#ECECEC;
    font-size:14pt;
}
```

- And assign a class to a node programmatically like so:

```
var myDiv = document.getElementById('myDiv');
myDiv.className = "someClassName";
```

# Excellent DOM Reference

- http://msdn.microsoft.com/library
  - Web Development
    - HTML and CSS
      - HTML and DHTML Reference
        - Objects
          - body, div, button, a, etc...

# Activity

- Style your div with a CSS selector that uses the ID of the DIV
- Style your text node by programmatically assigning className
- Style your button by programmatically setting properties
- Use: http://msdn.microsoft.com/library to find the available style properties for each type of object

# Homework

- Write 2 more tests for Magicalltem and 2 more tests for ShoppingCart
- Modify the prototype of the Array class to add a "Contains" function and a "Remove" function - both will take the item to be found/removed as a parameter
  - Use the w3c reference here: http://www.w3schools.com/jsref/jsref_obj_array.asp
- Go through the DOM tutorial at: http://www.w3schools.com/HTMLDOM/default.asp