WEB150 WN17

Assignment - Final Presentation, Underscore.js

Ron Nims

### Underscore.js Open Source Javascript library

Website: **underscorejs.org**

Github: **http://github.com/jashkenas/underscore/**

The underscore.js library is a swiss army knife of helpful javascript utility functions. It is easy to deploy and does not depend on other javascript libraries. The library is maintained and supported by the folks at underscorejs.org.

**From underscorejs.org homepage -**

*“*[*Underscore*](http://github.com/jashkenas/underscore/) *is a JavaScript library that provides a whole mess of useful functional programming helpers without extending any built-in objects. It’s the answer to the question: “If I sit down in front of a blank HTML page, and want to start being productive immediately, what do I need?” … and the tie to go along with* [*jQuery*](http://jquery.com/)*'s tux and* [*Backbone*](http://backbonejs.org/)*'s suspenders.”*

The one feature that I am focusing on for this presentation is underscore’s template functionality. Using their templates, it is very easy to reference members of a javascript data object from within the html structure.

**template** \_.template(templateString, [settings])

Compiles JavaScript templates into functions that can be evaluated for rendering. Useful for rendering complicated bits of HTML from JSON data sources. Template functions can both interpolate values, using <%= … %>, as well as execute arbitrary JavaScript code, with <% … %>. If you wish to interpolate a value, and have it be HTML-escaped, use <%- … %>. When you evaluate a template function, pass in a **data** object that has properties corresponding to the template's free variables. The **settings** argument should be a hash containing any \_.templateSettings that should be overridden.

var compiled = \_.template("hello: <%= name %>");  
compiled({name: 'moe'});  
=> "hello: moe"  
  
var template = \_.template("<b><%- value %></b>");  
template({value: '<script>'});  
=> "<b>&lt;script&gt;</b>"