**HTML Templates with Handlebars.js**

**Making JavaScript code by template!**

**Table of Contents**

* HTML Templates
* Handlebars.js Template Engine
  + Setup and usage
* Creating Templates
  + Binding values to HTML
* Handlebars Expressions
  + Block and conditional
* Helpers

**HTML Templates – Overview**

* HTML templates or template engines:
  + Make the HTML dynamic
    - Allow programming logic inside the otherwise static HTML
  + Provide cleaner and more sustainable way to dynamically create DOM elements
* HTML templates are many:
  + Basic: Handlebars.js, mustache.js, underscore.js, jQuery templates
  + Advanced: KendoUI, AngularJS

**Handlebars.js Overview (Setup and basics)**

* Handlebars.js is an open-source template engine
  + Provides both **run-time and compiled templates** for rendering HTML
  + Allows **one-way data-binding** of HTML elements to JavaScript objects
  + Produces HTML code based on **a template** and a given **JavaScript object**

**Handlebars.js: Setup and Usage**

* To use Handlebars follow the steps:
  + Acquire handlerbars.js:
    - From the site at  
      <http://handlebarsjs.com/>
    - Using bower:

$ bower install handlebars

* + - Src the script in the HTML file:

<script src="path/to/handlebars.js"></script>

* (cont.) To use Handlebars follow the steps:
  + Create a template
  + In a script tag with invalid type attribute

<div class='post'>

<h1 class="post-title">{{title}}</h1>

<p class="post-content">{{{content}}}</p>

</div>

* Render the template with JavaScript:

var post = {title: '…', content: '…'},

htmlTemplate = postTemplateNode.innerHTML,

postTemplate = Handlebars.compile(htmlTemplate),

postNode.innerHTML = postTemplate(post);

***Setup and Usage - Live Demo***

**Creating HTML Templates with Handlebars.js**

**(Start from the beginning)**

**Templates with Handlebars.js**

* HTML templates act much like

string.Format() in **C#** and **Java** and

stringWithFormat: @"…" in **Objective-C**

* + Put placeholders within a template string, and replace these placeholders with values
* Handlebars.js marks placeholders with **double curly brackets** {{value}}
  + When rendered, **the placeholders** between the curly brackets are **replaced with the corresponding value**

**Handlebars.js Templates Example**

* Creating the Template

<div class='post'>

<h1 class="post-title">{{title}}</h1>

<p class="post-content">{{content}}</p>

</div>

* Create the **JavaScript object** and render the template to **produce pure HTML** code

var post = { title: ' ... ', content: ' ... ' },

tmplStr = document.getElementById('...').innerHTML

template = Handlebars.compile(tmplStr);

domNode.innerHTML = template(post);

***Creating Handlebars Templates - Live Demo***

**HTML Escaping**

* Handlebars.js escapes all values before rendering them
* If the value should not be escaped there are two possibilities:
  + Use **triple curly brackets** in the template string:

{{{value}}}

* Mark in the value **not to be escaped**

var post = { title: '…',

content: new Handlebars.SafeString('…') *// Mostly used with helpers*

};

***HTML Escaping - Live Demo***

**Handlebars Expressions - Blocks and Conditionals**

**Handlebars Expressions**

* Handlebars.js supports expressions within the templates
  + Block expressions
    - For iterating over a collection of elements
  + Conditional expressions

**Block Expressions**

* Block expressions are used to iterate over a collection of objects (like array)
  + Created using {{#collection}} and {{/collection}}
    - Everything between will be evaluated for each object in the collection

<ul class="categories-list">

{{#categories}}

<li class="category-item">

<a href="#/categories/{{id}}">{{name}}</a>

*//The id and the name of every object*

</li>

{{/categories}}

</ul>

***Block Expressions - Live Demo***

**Conditional Expressions**

* Render code only if a condition is fulfulled
  + Using {{#if condition}} {{/if}}   
    or {{unless condition}} {{/unless}}

<h1>Posts</h1>

<ul class="posts">

{{#posts}}

<li>

<strong>s

<a href="#/posts/{{id}}">{{title}}</a>

</strong>

{{#if author}}

*//If author is false-like value, this code will not be rendered*

<span class="author">by {{author}}</span>

{{/if}}

</li>

{{/posts}}

</ul>

**Handlebars.js Helpers - Helping you even more!**

* Handlebars.js helpers are like functions that do predefined job
* Build-in helpers:
  + {{#each collection}} {{/each}}
    - Much like block expressions but with more control like

{{@index}} and {{this}}

* + {{#with obj}} {{/with}}
    - Used to shorten the path of property names
    - Write {{prop}} instead of {{obj.prop}}

***Handlebars.js Helpers - Live Demo***

**Registering Custom Helpers**

* Handlebars.js supports extension of the   
  built-in helpers
  + Using Handlebars.registerHelper(…);

Handlebars.registerHelper ('tags', function(tags){

var html = '<ul class="tags-list">',

tag;

for(tag in tags){

html += '<li class="tag">' + tag + '</li>';

}

*// SafeString is mandatory here; Otherwise the result will be escaped*

return new Handlebars.SafeString(html);

});

# *Registering Handlebars.js Helpers - Live Demo*