

Que Es Handlebars.Js Y Como Usarlo?

Anteriormente habíamos compartido [5 Herramientas De Desarrollo Web Que Deberias Conocer](#), entre estas herramientas se encuentra **Handlebars.Js**. En el presente articulo hablaremos mas en profundidad de **Handlebars.Js**, para que sirve, y dar un ejemplo practico de como usar esta herramienta de desarrollo web.



Primeros Pasos

Para empezar, vamos a ir a la página web de [Handlebars](#) y descargamos el archivo del código fuente, **handlebars.js**. Coloque el archivo en una carpeta adecuada de su proyecto. Enlace el archivo de los documentos HTML. Usted puede añadir el enlace dentro de la etiqueta del **head** o antes de la **<body>**.

```
[js]                                &amp;lt;script  
src=&quot;js/handlebars.js&quot;&amp;gt;&lt;/script&gt;  [/js]
```

Como buena alternativa, podemos añadir el CDN de esta libreria:

```
[js]                                &amp;lt;script  
src=&quot;//cdnjs.cloudflare.com/ajax/libs/handlebars.  
js/1.3.0/handlebars.js&quot;&amp;gt;  
&lt;/script&gt;  
[/js]
```

Una vez incluida podemos empezar a diseñar la plantilla.

Datos Básicos

Vamos a añadir los siguientes datos: nombre, edad, y el origen de la persona.

```
[js]
```

```
var dataSource = {  
  &quot;name&quot;: &quot;Joe Bloggs&quot;,  
  &quot;age&quot;: ,  
  &quot;19&quot;,  
  &quot;from&quot;: &quot;United Kingdom&quot;;  
}
```

```
[/js]
```

Estos datos sirven como ejemplo. Como se ha mencionado, puede recuperar formas similares de datos de la mayoría de las aplicaciones web que proporcionan una API abierta como **Twitter, Instagram, Flickr, Dribbble y Forrst** – aunque usted obtendrá mayores líneas de datos que lo que hemos mostrado anteriormente. Como alternativa, puede intentar usar un [generador de JSON](#) para generar algunos datos aleatorios.

La Plantilla

Una vez que tenemos los datos en nuestras manos, podemos **crear la plantilla** para colocar estos datos. Vamos a echar un vistazo a el siguiente código.:

```
[html]
```

```
&lt;div id=&quot;content&quot;  
class=&quot;content-wrap&quot;&gt;  
&lt;script id=&quot;template&quot;  
type=&quot;text/x-handlebars -
```

```
template&quot;&gt;&lt;p&gt;Hi, I'm {{name}}. I'm {{age}}, and  
I'm from {{from}}&lt;/p&gt;&lt;/script&gt;&lt;/div&gt;</html>
```

La Plantilla con **Handlebars** se establece dentro de una etiqueta de **script** de forma especial, **type: text/x-handlebars-template**, y preferiblemente también con un ID, porque la selección de un elemento con ID en JavaScript técnicamente es más sencillo y rápido que utilizar una clase.

Si bien cada uno de los datos se declara entre llaves dobles rizadas, `{{...}}`; Esto también se conoce como expresiones **Handlebars**.

Sin embargo, antes de que podamos ver el resultado en el navegador que tenemos que **compilar estos códigos**, combinar los datos con la plantilla.

Compilando

Vamos a almacenar la plantilla en una variable de **JavaScript**, de la siguiente forma:

```
[js]var template = $('#template').html(); [/js]
```

A continuación, ponemos la variable de plantilla en **Handlebars.compile()** para compilar la plantilla:

```
[js]var compile = Handlebars.compile(template);[/js]
```

Lo más importante en el código anterior es el nombre de la variable, `compile`. Vamos a utilizar junto con nuestros datos para generar el resultado final, del siguiente modo:

```
[js]var result = compile(dataSource);[/js]
```

Por último lo ponemos en `#content` usando el método `.html()` de jQuery, justo así:

```
[js]$('#content').html(result); [/js]
```

Esto nos dará el siguiente resultado en los navegadores:



Añadiendo HTML A Los Datos

A veces, nuestros datos podrían contener etiquetas **HTML**, por ejemplo:

```
[js]
var dataSource = {
  &&quot;name&&quot;;:
  &&quot;&&lt;em&&gt;Joe
  Bloggs&&lt;/em&&gt;&&quot;;,
  &&quot;age&&quot;;:
  &&quot;19&&quot;;,
  &&quot;from&&quot;;:      &&quot;United
  Kingdom&&quot;;
}

[/js]
```

Por defecto, Handlebars omitirá estas etiquetas. Ellos a su vez la etiqueta en su entidad, `<` se convertirá en `<`; mientras que `>` se convertirá en `>`. Así que vamos a obtener un resultado inesperado, como se muestra a continuación:



Para evitar que **Handlebars** omita la conversión de las etiquetas **HTML**, debemos utilizar llaves triples `{{{...}}}` para declarar los datos, así:

```
[js] Hi, My name is {{{name}}}. ... [/js]
```

Ahí lo tienes! La etiqueta ahora se procesa de forma correcta; el nombre se muestra en cursiva.



Ayuda Condicionales

Handlebars tiene soporte para funciones condicionales. Esta característica es un complemento exclusivo de **Handlebars**, no disponibles en Mustache. El ayudante condicional es útil para evitar la visualización de datos si el valor de datos está vacía. Como ejemplo, vamos a eliminar el valor de nuestros datos de age.

```
[js]
```

```
var dataSource = {  
  &&&quot;name&&&quot;;:      &&&quot;Joe  
  Bloggs&&&quot;;,  
  &&&quot;age&&&quot;;: &&&quot;&&&quot;;,  
  &&&quot;from&&&quot;;:      &&&quot;United  
  Kingdom&&&quot;;  
}
```

```
[/js]
```

Utilizamos el ayudante condicional de la siguiente manera. A medida que el valor de la edad no está presente, no vamos a mostrar la línea que dice **I'm {{{age}}}**,

```
[js] {{#if age}}I'm {{{age}}},{{/if}} [/js]
```

En los navegadores, se representará la línea de arriba.



Además, **Handlebars** también tomara a los datos como vacío si el valor se especifica explícitamente con: **undefined** o **false**.

Bucles ó Ciclos

Handlebars también soporta Bucles ó Ciclos. Como cualquier otro lenguaje de programación, se utiliza para iterar una serie de objetos. En este momento sólo tenemos un objeto que contiene tres líneas de datos . Vamos a ampliar nuestro ejemplo con dos objetos más, justo así:

```
[js]
```

```
var data = [{
  &quot;name&quot;      : &quot;Joe
Bloggs&quot;,
  &quot;age&quot;      :
&quot;19&quot;,
  &quot;from&quot;      : &quot;United
Kingdom&quot;,
}, {
  &quot;name&quot;      : &quot;Jane
Doe&quot;,
  &quot;age&quot;      :
&quot;21&quot;,
  &quot;from&quot;      : &quot;United
State&quot;,
}, {
  &quot;name&quot;      : &quot;John
Doe&quot;,
  &quot;age&quot;      :
&quot;20&quot;,
  &quot;from&quot;      : &quot;United
Nation&quot;,
}];
```

```
[/js]
```

Como ahora tenemos más de un objeto . Nuestra plantilla actual ya no funcionará para representar estos datos en la plantilla. En este caso , debemos utilizar **Handlebars Loop**, envuelva

nuestra plantilla dentro de `{{#each}} ... {{/each}}`. También podemos cambiar la plantilla si es necesario.

En este ejemplo , vamos a mostrar estos datos en una lista:

[js]

```
&amp;lt;script id=&amp;quot;template&amp;quot;
type=&amp;quot;text/x-handlebars-
template&amp;quot;&amp;gt;
&amp;lt;ul&amp;gt;
{{#each this}}
&amp;lt;li&amp;gt;Hi, My name is {{{name}}}. {{#if
age}}I&amp;#039;m {{age}},{{/if}} and
I&amp;#039;m from {{from}}&amp;lt;/li&amp;gt;
{{/each}}
&amp;lt;/ul&amp;gt;
&amp;lt;/script&amp;gt;
```

[/js]

Este es el resultado que vamos a ver en los navegadores.



Ejemplo Practico

Ahora vamos a implementar esta librería en un ejemplo real. Esta vez queremos mostrar un perfil de Forrst, **un centro para los diseñadores y desarrolladores**. Forrst proporciona un método simple para recuperar sus datos. En nuestro caso, podemos utilizar `{username}` para recuperar un perfil de usuario. Vamos a conseguir algo similar a estos datos a continuación.

Nota: los datos reales son demasiadamente largos. Así que omitiremos algunos de ellos para que sea más corto.

[js]

```

var forrstProfile = {
&quot;id&quot;: 24606,
&quot;username&quot;:
&quot;jimmyliu&quot;,
&quot;name&quot;:      &quot;Jimmy
Liu&quot;,
&quot;url&quot;:
&quot;http://forrst.com/people/jimmyliu&quot;,
&quot;likes&quot;:
&quot;11&quot;,
&quot;followers&quot;:
&quot;10&quot;,
&quot;following&quot;:
&quot;2&quot;,
&quot;photos&quot;: {
&quot;medium_url&quot;:
&quot;https://secure.gravatar.com/avatar/3151a9294
608c3143551aa265f00bf71.jpg?s=75&d=https://forrs
t.com/assets/images/default_75.jpg&quot;,
&quot;small_url&quot;:
&quot;https://secure.gravatar.com/avatar/3151a9294
608c3143551aa265f00bf71.jpg?s=45&d=https://forrs
t.com/assets/images/default_45.jpg&quot;,
&quot;thumb_url&quot;:
&quot;https://secure.gravatar.com/avatar/3151a9294
608c3143551aa265f00bf71.jpg?s=25&d=https://forrs
t.com/assets/images/default_25.jpg&quot;
},
&quot;bio&quot;:
&quot;&lt;p&gt;A graphic and web
designer based in Cupertino, California. Follow me on Twitter
&lt;a
href=&quot;http://twitter.com/jimmyliu&quot;
&gt;&lt;/a&gt;&lt;a
href=&quot;/people/jimmyliu&quot;
&gt;@jimmyliu&lt;/a&gt;&lt;/p&gt;

```

```
;gt;\n&amp;&amp;quot;;,  
&amp;&amp;quot;;is_a&amp;&amp;quot;;:    &amp;&amp;quot;;developer  
&amp;&amp;&amp; designer&amp;&amp;quot;;,  
};
```

```
[/js]
```

Ahora pondremos los datos en una plantilla:

```
[html]
```

```
&amp;&amp;lt;div      id=&amp;&amp;quot;;forrst&amp;&amp;quot;;  
class=&amp;&amp;quot;;forrst-profile&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;script      id=&amp;&amp;quot;;forrst-profile-  
template&amp;&amp;quot;;  type=&amp;&amp;quot;;text/x-handlebars-  
template&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;div  
class=&amp;&amp;quot;;profile&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;figure  
class=&amp;&amp;quot;;avatar&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;img  
src=&amp;&amp;quot;;{{photos.medium_url}}&amp;&amp;quot;;  
alt=&amp;&amp;quot;;&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;/figure&amp;&amp;gt;  
&amp;&amp;lt;div  
class=&amp;&amp;quot;;person&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;h3      class=&amp;&amp;quot;;person-  
name&amp;&amp;quot;;&amp;&amp;gt;&amp;&amp;lt;a  
href=&amp;&amp;quot;;{{url}}&amp;&amp;quot;;&amp;&amp;gt;{{name}}&  
mp;&amp;lt;/a&amp;&amp;gt;&amp;&amp;lt;/h3&amp;&amp;gt;  
&amp;&amp;lt;div      class=&amp;&amp;quot;;person-  
bio&amp;&amp;quot;;&amp;&amp;gt;{{{bio}}}&amp;&amp;lt;/div&amp;&  
;gt;  
&amp;&amp;lt;/div&amp;&amp;gt;  
&amp;&amp;lt;/div&amp;&amp;gt;  
&amp;&amp;lt;div      class=&amp;&amp;quot;;social-  
count&amp;&amp;quot;;&amp;&amp;gt;  
&amp;&amp;lt;div  class=&amp;&amp;quot;;social-item  social-
```

```

posts&quot;&gt;
&lt;div
class=&quot;heading&quot;&gt;Posts&
&lt;/div&gt;
&lt;div
class=&quot;counts&quot;&gt;{{posts}}&
&lt;/div&gt;
&lt;/div&gt;
&lt;div class=&quot;social-item social-
followers&quot;&gt;
&lt;div
class=&quot;heading&quot;&gt;Followers
&lt;/div&gt;
&lt;div
class=&quot;counts&quot;&gt;{{follower
s}}&lt;/div&gt;
&lt;/div&gt;
&lt;div class=&quot;social-item social-
following&quot;&gt;
&lt;div
class=&quot;heading&quot;&gt;Following
&lt;/div&gt;
&lt;div
class=&quot;counts&quot;&gt;{{followin
g}}&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/script&gt;
&lt;/div&gt;

[/html]

```

Ahora compilaremos:

```
[js]
```

```

var template = $('#forrst-profile-template').html();
var compile = Handlebars.compile(template);

```

```
var result = compile(forrstProfile);  
$('#forrst').html(result);  
  
[/js]
```

Con un par de líneas de CSS, podemos lograr un resultado bastante presentable. En el siguiente enlace encontraras la plantilla funcional.