

ceviche.js

Apéndice B: Bibliotecas de terceros

Cuando nos metemos al desarrollo de un sitio web, es común encontrarse con necesidades que pueden ser cubiertas fácilmente por código de otras personas. De esta forma, podemos ahorrar mucho tiempo y confiar que nuestro código funcione sobre el código de otros. Estas son algunas bibliotecas útiles en cualquier desarrollo web.

jQuery

[jQuery](#) es la biblioteca más popular de JavaScript y resuelve uno de los más grandes problemas en el desarrollo web: El código *cross-browser*. Hace muchos años, se tenía que crear dos versiones del mismo código: una para Netscape y otra para Internet Explorer. Cuando Netscape desapareció y apareció Firefox, se dio el mismo caso, una vez más con Internet Explorer del otro lado. Si a eso le sumamos otros navegadores, como Opera o Safari (para Mac OS), el código crece rápidamente.

jQuery tiene [documentación con ejemplos](#), dividida en las siguientes secciones:

- **Ajax:** Funciones relacionadas a operaciones asíncronas
- **Attributes:** Métodos que manejan atributos de elementos
- **Callbacks:** Objeto para manejar callbacks (tal y como vimos en [el capítulo 2](#))
- **Core:** Funciones relacionadas al uso de jQuery
- **CSS:** Métodos que manejan estilos de elementos
- **Data:** Métodos que manejan datos de elementos (utilizando la propiedad `dataset`)
- **Deferred Object:** Objeto que permite manejar promesas
- **Deprecated:** Funciones y métodos que serán eliminados
- **Dimensions:** Métodos que manejan dimensiones de elementos (alto y

ancho)

- **Effects:** Animaciones simples y de bajo nivel
- **Events:** Métodos para manejar eventos del DOM y eventos propios
- **Forms:** Métodos relacionados al manejo de formularios
- **Internals:** Propiedades relacionadas a cada instancia de jQuery
- **Manipulation:** Métodos para manipular elementos (agregar y eliminar elementos)
- **Miscellaneous:** Funciones y métodos accesorios
- **Offset:** Métodos utilizados para obtener la posición de elementos dentro del documento
- **Properties:** Propiedades globales de jQuery
- **Removed:** Funciones y métodos que han sido eliminadas
- **Selectors:** Selectores utilizados para obtener elementos en jQuery. Estos selectores son, en su mayoría, CSS, pero existen selectores propios de jQuery.
- **Traversing:** Métodos utilizados para navegar entre elementos del DOM
- **Utilities:** Funciones utilitarias

Zepto

[Zepto](#) es una biblioteca similar a jQuery, aunque de menor tamaño. Zepto le da soporte solo a navegadores modernos, como:

- Safari 6+ (Mac)
- Chrome 30+ (Windows, Mac, Android, iOS, Linux, Chrome OS)
- Firefox 24+ (Windows, Mac, Android, Linux, Firefox OS)
- iOS 5+ Safari
- Android 2.3+ Browser
- Internet Explorer 10+ (Windows, Windows Phone)

Y soporta parcialmente:

- Safari 3+ (iOS)
- Chrome <30
- Firefox 4+
- Safari <6 (Mac)
- Android Browser 2.2

- Opera 10+
- webOS 1.4.5+ Browser
- BlackBerry Tablet OS 1.0.7+ Browser
- Amazon Silk 1.0+
- Otros navegadores basados en WebKit

Esto difiere de jQuery, que soporta a todos los navegadores posibles en su rama 1.x (la versión 2.0 de jQuery en adelante soporta solo navegadores modernos).

Una característica importante de Zepto es que la biblioteca por defecto contiene ciertos módulos: Core, Events, Ajax, Form, IE. Si se desea tener la biblioteca con todos los módulos, se debe [construir la biblioteca desde el código fuente](#).

Underscore.js

[Underscore](#) es una biblioteca que contiene métodos necesarios para realizar operaciones con arreglos, objetos, colecciones (arreglos u objetos) y funciones. Así mismo, contiene otras funciones utilitarias.

Underscore tiene su [código fuente comentado](#), información muy útil para aprender cómo mejorar nuestro propio código.

Dentro de los métodos más populares dentro de Underscore, se encuentran:

- `each`: Itera por cada valor de un arreglo, o por cada valor de cada par llave-valor de un objeto
- `map`: Similar a `each`, pero devuelve un arreglo con los valores devueltos por una función pasada como parámetro
- `compact`: Devuelve una copia de un arreglo con los elementos que no sean *falsy*
- `uniq`: Devuelve una copia de un arreglo sin los elementos repetidos
- `bind`: Cambia el contexto de una función, y define parámetros. Similar a `Function.prototype.call`, pero no ejecuta la función (mientras que `Function.prototype.call` sí lo hace)
- `values`: Devuelve un arreglo con los valores de un objeto
- `extend`: Copia las propiedades de un objeto a otro
- `template`: Permite crear plantillas utilizando una cadena de texto,

devolviendo una función que, al ser ejecutada, reemplazará los valores de la plantilla por los que sean pasados como parámetros.

Moment.js

Moment permite manejar fechas de una forma mucho más completa que lo ofrecido por defecto. No solo permite mostrar una fecha en diferentes formatos (que puede ir desde `31/12/2014` hasta `miércoles, 31 de diciembre de 2014 0:00`), si no que permite mostrar fechas en diferentes idiomas (79 idiomas hasta ahora) y permite realizar operaciones entre fechas.

Adicionalmente, existe una biblioteca llamada **Moment Timezone** que permite manejar fechas entre zonas horarias, así como darles formato y realizar todas las operaciones disponibles en Moment.js.

[← 1. Apéndice A: Referencia DOM](#)

[Acerca de](#) | [Disponible en Amazon.com](#)