

# Laboratorio #5: JQuery

## JQuery

JQuery es una librería multi-navegador de JavaScript diseñada para simplificar la labor de programar comportamiento en forma de scripts del lado del cliente. Fue liberada en Enero del 2006 en el BarCamp NY por John Resig. Es actualmente desarrollado por un equipo liderado por Dave Methvin. Utilizado por más del 65% de los 10000 sitios web más visitados, JQuery es el *framework* JavaScript más popular en uso hoy en día.

JQuery es una librería de código abierto, gratuita, bajo la licencia MIT<sup>1</sup>, y pensada para hacer más sencilla la manipulación de los elementos del DOM, crear animaciones, manejar eventos y desarrollar aplicaciones con el enfoque AJAX. JQuery igualmente provee la capacidad para desarrolladores de crear plug-ins sobre la librería. [1]

Es relevante tener en cuenta que existen muchas otras librerías que proveen funcionalidades JavaScript de distintos tipos, no obstante por la difusión que tiene JQuery y por la gran versatilidad de la herramienta es la de escogencia para el laboratorio. Otras posibles bibliotecas conocidas son Twitter Bootstrap, Prototype JavaScript Framework, Yahoo! User Interface library entre muchas otras.

JQuery tiene proyectos asociados bastante útiles para tareas más específicas aunque igualmente comunes a desarrollos de código del lado del cliente, como son JQuery User Interface (herramientas de interacción con interfaces de usuario, efectos, widgets y temas) [2], JQuery Mobile (provee de una interfaz móvil basada en HTML5 que permita desarrollar código único para aplicaciones móviles compatible con los distintos navegadores provistos en los sistemas operativos móviles modernos) [3], QUnit (herramienta para pruebas de código JavaScript) [4] y Sizzle, el motor selector basado en selectores CSS [5].

## Características

1. **Selección de elementos DOM utilizando el motor selector multi-navegador Sizzle**
2. **DOM y modificación por recorrido:** La naturaleza de los selectores de JQuery y la aplicación de funciones genera que la selección y aplicación se realice en forma de recorridos sobre los elementos.
3. **Manipulación del DOM basada en selectores CSS**
4. **Manejo de eventos**
5. **Manejo de eventos y animaciones**
6. **Manejo de AJAX multi-navegador**
7. **Extensibilidad a través de plug-ins**

<sup>1</sup> <http://opensource.org/licenses/MIT>

8. **Utilidades adicionales:** JQuery provee de ciertas utilidades adicionales para los programadores como información sobre los agentes de usuario, detección de características, entre otras.
9. **Funciones que proveen compatibilidad a JQuery con sus equivalentes en JavaScript:** Los objetos manejados por los selectores de JQuery no son los objetos DOM manejados por JavaScript. Por esto la biblioteca provee funciones que cumplen con las mismas funciones que las rutinas definidas en JavaScript, con la excepción de algunas más antiguas.
10. **JQuery es multi-navegador**

[1]

### Uso de JQuery

JQuery maneja algunas estructuras no clásicas de JavaScript las cuales en buena parte, proveen los principales atractivos de la librería.

En general JQuery tiene el comportamiento usual de JavaScript, con la excepción que la mayoría de las acciones se basan en métodos del objeto JQuery, que ya definiremos, que simplifican el manejo y que, basados en el DOM y modificación por recorrido minimizan la cantidad de código escrito.

#### **El objeto JQuery**

Un objeto JQuery contiene una colección de elementos DOM que fueron creados de strings HTML o de selecciones de un documento. El objeto JQuery en sí mismo es semejante a un arreglo, tiene la propiedad *length* y los elementos en el arreglo pueden ser accedidos mediante su índice, sin embargo no es un objeto de arreglo de JavaScript explícitamente por lo cual no provee de todas las funciones que tendría disponible un objeto de arreglo de JavaScript.

El objeto JQuery puede ser obtenido por medio de su constructor o por medio de su alias más conocido, el método \$.

El objeto JQuery es interesante y útil por la forma en la cual la librería lo maneja y lo cual permite el encadenamiento de métodos sobre el mismo objeto JQuery. La mayoría de los métodos del objeto JQuery retornan el mismo objeto JQuery, lo cual genera la técnica de uso clásica de JQuery

```
$('#select#carmakes').append($('gen"));
```

[6]

#### **Selectores:**

JQuery, al aplicar el motor Sizzle, utiliza selectores CSS para aplicar la construcción del objeto JQuery. No obstante, además de mantenerse al día

con los selectores de los distintos niveles CSS, crean algunos grupos selectores que es común requerirlos en el desarrollo web estándar. Algunos ejemplos de estos selectores son `:animated`, `:checked`, `:empty`.

### El método \$

El método constructor del objeto JQuery es el fundamento de una considerable parte de la librería. El método `$`, siendo un *factory method*, según la cantidad de parámetros, el tipo de parámetros o si quiera si se le colocan parámetros.

El caso común para el uso de esta constructora es cuando el parámetro es un `string` selector al estilo CSS. En dicho caso el objeto JQuery retornado son aquellos elementos que correspondan con el selector indicado.

```
$('#id, .class, tagName')
```

Para el caso que se coloquen como parámetros objetos DOM o planos, el constructor coloca esos elementos DOM dentro de la colección del objeto JQuery.

```
$([document.createElement('p'), document.createElement('div')])
```

Cuando el parámetro es un `string` HTML la colección del objeto JQuery consta del objeto DOM representante de ese `string`. Es importante saber que sólo se puede realizar conversión de documentos HTML, no es el caso de documentos XML.

```
$('#id, .class, tagName')
```

### Métodos

JQuery en general se maneja con eventos aplicados al objeto JQuery o alguna de las estructuras definidas por JQuery. La librería provee diversos métodos pensando en diversas actividades recurrentes en el desarrollo del lado del cliente de JavaScript. Los mismos se encuentran bien documentados y ejemplificados en la web. Queda de parte del lector documentarse sobre estas.

[7]

### Funciones Callback

Las funciones *callback* son funciones (JavaScript) pasadas como parámetro a algún método para que este las ejecute en algún punto conveniente. En particular en JQuery esto viene a representar la asignación de manejadores de evento (*event handlers*) y es común encontrarse con esta estructura.

Algunas funciones *callback* proveen argumentos y un contexto. Para el ejemplo siguiente utilizaremos un *event handler* cuyo argumento es el objeto del evento. El contexto será aplicar el *event handler* al body del documento.

```
$( "body" ).click(function( event ) {  
    console.log( "clicked: " + event.target );  
});
```

Igualmente, algunas funciones *callback* se espera que tengan ciertos retornos. Esto debe verificarse en la documentación de cada método.

[6]

### Otros elementos JQuery

JQuery es una librería bastante grande y versátil. La misma define distintos objetos con objetivos un poco más particulares las cuales quedan a su interés comprender y estudiar.

## Referencias

- [1] <http://en.wikipedia.org/wiki/JQuery>
- [2] <http://jqueryui.com/>
- [3] <http://jquerymobile.com/>
- [4] <http://qunitjs.com/>
- [5] <http://sizzlejs.com/>
- [6] <http://api.jquery.com/Types/>  
<http://api.jquery.com/>