

UD 10. INTRODUCCIÓN A JQUERY

Desarrollo web en entorno cliente
CFGS DAW

Sergio García Barea
sergio.garcia@ceedcv.es

2019/2020

Versión:191010.0934


Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

ÍNDICE DE CONTENIDO

1. Introducción.....	3
2. Incluyendo la biblioteca jQuery.....	3
3. Función \$().....	4
4. Código inicial en jQuery: \$(document).ready.....	5
5. Algunas funciones útiles.....	5
5.1 CSS.....	5
5.2 Filtros.....	5
5.3 Modificando DOM.....	6
5.4 Manejando eventos.....	6
5.5 Animaciones en jQuery: mostrar/ocultar.....	7
5.6 Animaciones en jQuery: modificar propiedades CSS.....	7
6. AJAX en JQuery.....	8
7. Material adicional.....	8
8. Bibliografía.....	9

UD10. INTRODUCCIÓN A JQUERY

1. INTRODUCCIÓN

La biblioteca jQuery (página oficial <https://jquery.com/>) es una biblioteca que permite:

- Simplificar la forma de interactuar con elementos XHTML.
- Manipular el DOM la página.
- Trabajar con conjuntos (de forma similar a lenguajes como Python).
- Realizar algunas animaciones de elementos.
- Utilizar la técnica AJAX.

Esta biblioteca es actualmente una de las más extendidas en el desarrollo web mediante Javascript.

2. INCLUYENDO LA BIBLIOTECA JQUERY

La biblioteca jQuery puede descargarse e incluirse localmente o incluirse online (se incluye un fichero Javascript alojado en un servidor externo).

Las ventajas de incluirse online son:

- Las inclusiones online suelen ser de sitios muy concretos y los proveedores de Internet suelen tenerlas almacenadas en cache para acceso rápido.
- Las inclusiones online te permiten tener siempre la versión actualizada sin realizar cambios.

Las desventajas de incluirse online son:

- Si la maquina cliente que quiere acceder a ella, no puede tener acceso (no hay Internet, un firewall le bloquea el acceso a la página, la página esta caída, etc.) no podrá ejecutar la aplicación Web.
- Si un atacante modifica el sitio web donde esta alojada la librería, podría troyanizarla creando un problema de seguridad.

Para incluirla localmente, solo debemos descargarla e incluirla como un fichero ".js" más.

Para incluirla online, podemos coger la referencia de sitios como <http://code.jquery.com/> obteniendo un código similar a este

```
<script src="http://code.jquery.com/jquery-3.1.1.slim.min.js"
integrity="sha256-/S1rNqv8h6QGKDuNoLGA4iret+kyesCkHGzVUUV0shc="
crossorigin="anonymous"></script>
```

3. FUNCIÓN \$()

La función `$()` es el alma del uso de jQuery y el primer concepto básico.

Básicamente la función `$()` es una función que recibe un nombre de elemento XHTML o CSS y devuelve el elemento o conjunto de elementos que casan por lo pedido.

Ejemplo

```
$("#prueba"); // Devolverá el elemento con id="prueba"
$(".alumno"); // Devolverá una conjunto de elementos con class="alumno"
$("div"); // Devolverá una conjunto de elementos con todos los elementos div
```

Al elemento o conjunto de elementos obtenidos, se les pueden aplicar las distintas funciones existentes en la biblioteca.

Ejemplo

```
$(".alumno").removeClass("alumno").addClass("exAlumno");
//Hace que los elementos de la clase alumno dejen de ser parte de esa clase. La
funcion removeClass devuelve los elementos afectados como conjunto y a este se
le aplica assClass y en la práctica pasan a ser todos esos elementos de la clase
"exAlumno."
$(".exAlumno").slideToggle("slow");
// Aplica a los elementos de la clase exAlumno la posibilidad de
ocultarse/mostrarse con una animación
```

Para conocer las funciones disponibles y sus parámetros, entre otras fuentes de información podéis usar la API de jQuery <http://api.jquery.com> que es muy completa y contiene ejemplos prácticos de los elementos documentados.

Por ejemplo, las entradas concretas de las funciones utilizadas anteriormente:

- `addClass` <http://api.jquery.com/addClass/>
- `removeClass` <http://api.jquery.com/removeClass/>
- `slideToggle` <http://api.jquery.com/slidetoggle/>

La función `$()` internamente es un alias del objeto jQuery. Si lo deseamos podemos crear nuestro propio alias que la sustituya de la siguiente forma.

Ejemplo

```
var miAlias=jQuery;
mialias.ready(function() {
    //Aquí van todas las acciones del documento.
});
```

El uso de alias es útil cuando hay conflictos con otras bibliotecas Javascript que utilicen el alias `$`.

Mas información en <http://learn.jquery.com/using-jquery-core/avoid-conflicts->

[other-libraries/](#)

4. CÓDIGO INICIAL EN JQUERY: \$(DOCUMENT).READY

Para comenzar a utilizar jQuery, debemos esperar a que el documento este listo. Ahí empezaremos a utilizar nuestro código.

Incluiremos nuestro código inicial en \$(document).ready (El equivalente a Main en otros lenguajes de programación).

Ejemplo

```
$(document).ready(function() {  
    //Aquí van todas las acciones del documento.  
});
```

5. ALGUNAS FUNCIONES ÚTILES

5.1 CSS

Seleccionado un elemento o conjunto de ellos, con la función css podemos modificar su código CSS.

De manera básica podemos acceder a cualquier propiedad (usando su nombre CSS) y leerla o modificarla.

- css(propiedad): devuelve el valor de dicha propiedad
- css(propiedad,valor): modifica el valor de dicha propiedad.

Ejemplo

```
var color = $("#miElemento").css("background-color"); // Obtiene el color  
$("#miElemento").css("background-color","red"); // Pone el fondo rojo
```

Mas información en <http://api.jquery.com/css/>

5.2 Filtros

Hemos visto que jQuery permite obtener conjuntos de elementos. A veces merece la pena aplicar filtros a los conjuntos de elementos obtenidos. El selector aplicado indicará que conjunto se desea recibir.

En <https://api.jquery.com/category/selectors/> tenéis los selectores disponibles. En el ejemplo, se usa :even que sirve para filtrar solo los impares. Otros tipicos suelen ser :first, :last, :odd, :hidde, :visible

Ejemplo

```
$("li").filter(":even").css("background-color","red");  
//Pone el fondo rojo rojo los li impares
```

Más información en <http://api.jquery.com/filter/>

5.3 Modificando DOM

La biblioteca jQuery nos permite entre otras cosas modificar el DOM de la página.

Ya hemos visto como acceder a los elementos DOM con `$()`. Ahora veremos algunas de las funciones más típicas para modificar su estructura:

- `html(nuevoContenido)`: permite modificar el contenido HTML de un elemento. Si se llama sin parámetros, simplemente devuelve el HTML actual del elemento.
 - <http://api.jquery.com/html/>
- `append(elemento)/prepend(elemento)`: añade un elemento como hijo del elemento actual al final o al principio.
 - <http://api.jquery.com/append/> y <http://api.jquery.com/prepend/>
- `empty()`: vacía el contenido HTML de un elemento.
 - <https://api.jquery.com/empty/>
- `remove()`: elimina el conjunto de elementos al que se le aplique
 - <https://api.jquery.com/remove/>

Ejemplo

```
miHTML=$("#miDiv").html();// Obtiene el HTML del elemento
$("#miDiv").html("<b>Hola</b>");// Modifica el HTML del elemento
$("p").empty(); // Vacía todos los elementos P
$("#miDiv").append("<p>Probando</p>");// Añade el elemento P a #miDiv
$("div").remove(); // Elimina todos los divs
```

Más información en <http://api.jquery.com/category/manipulation/dom-insertion-inside/> y en <https://api.jquery.com/category/manipulation/dom-removal/>

5.4 Manejando eventos

La biblioteca jQuery nos permite manejar eventos de los distintos elementos.

En esta página podéis ver una lista de eventos que pueden ser manejados.

<https://api.jquery.com/category/events/>

Aquí un ejemplo con click <https://api.jquery.com/click/>

Ejemplo

```
$( "#target" ).click(function() {
    alert( "Handler for .click() called." );
});
```

5.5 Animaciones en jQuery: mostrar/ocultar

Se pueden realizar animaciones en jQuery que permiten dotar de mas vistosidad a la página.

Hay dos elementos comunes a casi todas las animaciones:

- Se les puede pasar el tiempo que dura la animación en ms. Aparte de forma numérica, también se puede pasar la velocidad de la animación en modo texto (slow: 600ms, normal: 400ms, fast: 200ms).
- Se les puede incluir al final una función de callback (función que se ejecuta al finalizar la animación). Si el selector aplica a varios elementos, esta función se llamará una vez por cada elemento afectado.

Un ejemplo es la típica de usar hide() and show().

```
$('#elemento').show(3000); // Hace que se muestre un elemento oculto
// El tiempo indica cuanto dura en ms la animacion de mostrado
$('.clase').hide(4000);
```

Existen otros métodos similares como fadeIn() y fadeOut(). Como hide y show, pero se basan en aumentar/reducir la opacidad como animación.

```
$('.p').fadeIn();
```

Existe otro método, fadeTo que permite en vez de ocultar o mostrar ajustar la opacidad.

```
$('.p').fadeTo("normal", 0.6, function() {
    // función callback
});
```

Donde la velocidad es normal, la opacidad es 0.6 y se define una función de callback vacía.

5.6 Animaciones en jQuery: modificar propiedades CSS

En jQuery podemos cambiar propiedades CSS de elementos y que dichos cambios se muestren como animacion con la funcion animate.

Básicamente a animate se le pasan los nuevos valores de las propiedades del selector y opcionalmente otros parámetros como velocidad, función callback, etc.

```
$( ".block" ).animate({ "left": "-=50px" }, "slow" );
```

Anima los elementos de clase “block” moviéndolos 50 pixeles a la izquierda.

En la propia API de jQuery tenéis varios ejemplos interesantes con código.

<http://api.jquery.com/animate/>

6. AJAX EN JQUERY

La biblioteca jQuery nos permite de manera sencilla realizar las acciones que nos permite usar AJAX. Tenéis una descripción de los métodos para usar AJAX en <https://api.jquery.com/category/ajax/>

El método principal es ajax <https://api.jquery.com/jQuery.ajax/>

Ejemplo

```
$.ajax({
  method:"POST",
  url:"prueba.php",
  data:{ nombre: "Casimiro", apellido: "Buenavista" }
})
.done(function(msg) {
  alert("Recibo: " + msg);
})
.fail(function() {
  alert("Error");
})
.always(function() {
  alert("Trabajo realizado");
});
// Ajax que envia dos parametros a prueba.php por metodo POST y cuando recibe la
// respuesta la imprime con un alert. Si falla muestra "Error" y siempre muestra
// "Trabajo realizado".
```

También hay métodos alias de \$.ajax que nos permiten agilizar el trabajo, como por ejemplo <https://api.jquery.com/jQuery.getJSON/> que automáticamente parsea a JSON los datos recibidos.

🔊 Recordamos que también se puede parsear a mano JSON con JSON.stringify https://www.w3schools.com/js/js_json_stringify.asp y JSON.parse https://www.w3schools.com/js/js_json_parse.asp

7. MATERIAL ADICIONAL

[1] Fundamentos de jQuery https://librosweb.es/libro/fundamentos_jquery/

[2] Intro to AJAX (Udacity) <https://www.udacity.com/course/intro-to-ajax--ud110>

[3] Practicar jQuery W3Schools <https://www.w3schools.com/jquery/>

[4] Ejemplos básicos de jQuery

<https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4822881-agrega-gestores-de-eventos>

8. BIBLIOGRAFÍA

[1] Referencia Javascript

<http://www.w3schools.com/jsref/>

[2] API jQuery

<http://api.jquery.com>

[3] W3Schools jQuery

<https://www.w3schools.com/jquery/>