

DWEC

TEMA 6



JQUERY



ÍNDICE

- 6.1 Introducción
- 6.2 Incluir biblioteca JQuery
- 6.3 Función \$()
- 6.4 Funciones Útiles.
- 6.5 Ajax en JQuery

6.1 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 ha sido una de las más extendidas en el desarrollo web mediante Javascript, aunque está quedando en desuso por la irrupción de la programación reactiva (Vue, React, Angular, etc.).

6.2 INCLUIR 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 caché para acceso rápido.
- Las inclusiones online te permiten tener siempre la versión actualizada sin realizar cambios.

Las desventajas de incluirse online son:

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

6.2 INCLUIR BIBLIOTECA JQUERY

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-3.5.1.slim.js"  
integrity="sha256-DrT5NfxfbHvMHux31Lkhxg42LY6of8TaYyK50jnxRnM="  
crossorigin="anonymous"></script>
```

6.2 INCLUIR BIBLIOTECA JQUERY

En una aplicación de JavaScript, para comenzar a utilizar jQuery, debemos esperar a que el documento esté listo. Ahí empezaremos a utilizar nuestro código. Incluiremos nuestro código inicial en `$(document).ready` (El equivalente a Main en otros lenguajes de programación).

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

Para poder utilizar jQuery cuando estéis desarrollando en un entorno tipo NodeJS debéis incluir en el fichero principal de Javascript de vuestra aplicación las siguientes líneas. Estas deben ejecutarse al iniciar la aplicación.

```
var $ = require('jquery');  
window.jQuery = $;  
window.$ = $;
```

6.3 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 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.
```

6.4 FUNCIONES ÚTILES

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.

6.4 FUNCIONES ÚTILES

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:

```
let 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/>

6.4 FUNCIONES ÚTILES

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 qué conjunto se desea recibir.

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

6.4 FUNCIONES ÚTILES

Modificando DOM

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

Ya hemos visto cómo 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/>

6.4 FUNCIONES ÚTILES

Modificando DOM

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/>

<https://api.jquery.com/category/manipulation/dom-removal/>

6.4 FUNCIONES ÚTILES

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() {  
    console.log( "Handler for .click() called." );  
});
```

6.4 FUNCIONES ÚTILES

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() {  
    console.log( "Handler for .click() called." );  
});
```

6.4 FUNCIONES ÚTILES

Animaciones en jQuery: mostrar/ocultar

Se pueden realizar animaciones en jQuery que permiten dotar de más 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” (milisegundos). 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 usando hide() and show():

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

6.4 FUNCIONES ÚTILES

Animaciones en jQuery: mostrar/ocultar

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.

6.4 FUNCIONES ÚTILES

Animaciones en jQuery: modificar propiedades CSS

En jQuery podemos cambiar propiedades CSS de elementos y que dichos cambios se muestren como animación con la función “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” moviéndose 50 píxeles a la izquierda.

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

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

6.5 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");  
  });
```