

Tema 7

jQuery

JQuery

jQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones.

Creada inicialmente por John Resig, que permite:

- Simplificar la manera de interactuar con los documentos HTML.
- Manipular el árbol DOM.
- Manejar eventos.
- Desarrollar animaciones.
- Agregar interacción con la técnica AJAX a páginas web

JQuery

Ventajas:

- Tamaño relativamente reducido.
- Cómoda para los diseñadores web.
- Está probada y funciona.
- Es gratis.
- Hay una gran comunidad de desarrolladores.
- Complementos, complementos y más complementos.
- Software libre y de código abierto.

JQuery

Jquery es un archivo javascript que hay que vincularlo en la página web.

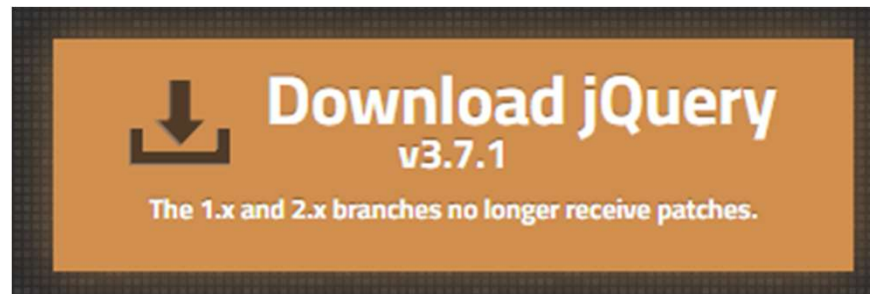
Dispone de varias opciones:

- Vincularlo en una CDN (Red de distribución de contenidos); es decir, otro sitio web que aloja el archivo jQuery y lo envía a todo el que lo solicita.
- Descargarlo localmente y añadirlo al sitio junto a las demás páginas web y archivos

Básicamente lo mismo que sucede con bootstrap, y al igual que con bootstrap es preferible siempre instalarlo localmente, por que ya hemos visto que si lo vinculas con el CDN es más simple pero dependes de que no haya problemas con la conexión u otros problemas que puedan ocurrir en el servidor donde está alojado.

jQuery

Para descargar el archivo visitamos <https://jquery.com> y descargar la última versión.



El archivo jQuery se ofrece en tres versiones:

- Minified: reducida.
- Uncompressed: sin comprimir.
- Slim: no tiene ajax ni eventos.

Nosotros nos descargaremos la versión minified.

jQuery

En primer lugar tenemos que enlazar el script jQuery a nuestro index, y es recomendado que se ponga al final del body pero antes de nuestro propio script para que nuestro script se ejecute después del script de jQuery.

Para poder realizar algo en nuestro script tendremos que esperar a que se cargue la librería jQuery y la estructura del DOM. Para ello se utiliza la función `ready()`.

```
JS miScript.js > ...  
1  $(document).ready(function() {  
2    |    console.log($("#div")[0]);  
3  });|
```

jQuery

Hemos visto en la imagen anterior la sintaxis:

```
$(“div”)[0]
```

Vamos a analizarlo:

La función dolar es la función principal de jQuery y nos permite acceder a los nodos del DOM. En este caso todos los nodos div que tenga el DOM, por eso hemos seleccionado solo una posición.

Probarlo!

jQuery

También podemos declarar la función de la siguiente forma:

```
$(function(){  
  console.log($(".div")[0])  
})
```

Aunque la anterior es más correcta, pero si probamos las dos funcionan.

jQuery

Selectores.

jQuery utiliza los selectores para acceder de forma más rápida y sencilla a un elemento:

- Selectores CSS.
- Selectores jQuery
- Selectores de formulario.

jQuery

Selectores CSS.

`$('*')`-> Selecciona todos los elementos.

`$('p')`-> Selecciona todos los `<p>` de la página.

`$('p','a')`-> Selecciona todos los `<p>` y todos los `<a>` del documento.

`$('p a')`-> Selecciona todos los `<a>` descendiente de `<p>`.

`$('li.nombreClase')`-> Selecciona todos los `` cuya clase sea 'nombreClase'.

`$('li>p')`-> Selecciona todos los `<p>` hijos directos de ``

`$('h1+p')`-> Selecciona todos los `<p>` inmediatamente precedidos por `<h1>`
hermano

jQuery

Selectores Css.

`$('p:has(b)')`-> Selecciona todos los `<p>` que contienen un elemento ``.

`$('.nombreClase')`-> Selecciona todos los elementos con la clase 'nombreClase'.

`$('#nombreID')`-> Selecciona todos los elementos con el ID.

`$('p:empty')`-> Selecciona todos los `<p>` que estén vacíos.

`$('img[alt]')`-> Selecciona todos los elementos `` que posean atributo alt

jQuery

Selectores CSS

`$(':button[id*=boton]')`->Selecciona todos los botones con atributo 'id' que contengan la palabra boton.

`$('a[href$=.pdf]')`->Selecciona todos los <a> cuyo atributo 'href' acabe en pdf.

`$('a[title^=.lr]')`->Selecciona todos los <a> cuyo atributo 'title' empiece en '.lr'

jQuery

Selectores jQuery.

`$('p:first')`-> Selecciona el primer elemento `<p>` de la página.

`$('p:last')`-> Selecciona el último elemento `<p>` de la página.

`$('li:first-child')`-> Selecciona todos los `` que son primeros hijos.

`$('li:only-child')`-> Selecciona todos los `` que sean hijos únicos.

`$('li:nth-child(3)')`-> Selecciona todos los `` que sean el tercer elemento de su lista

jQuery

Selectores jQuery

`$('tr:nth-child(odd)')`-> Selecciona todos los `<tr>` que sean impares.

`$('tr:nth-child(even)')`-> Selecciona todos los `<tr>` que sean pares.

`$('p:odd')`-> Selecciona los `<p>` impares.

`$('p:eq(1)')`-> Selecciona el segundo `<p>`. (Comienza por cero como si fuera un array)

`$('p:gt(1)')`-> Selecciona todos los `<p>`, excepto los dos primeros

`$('p:lt(2)')`-> Selecciona los dos primeros `<p>`

`$('p:contains(texto)')`-> Selecciona los `<p>` que contengan el texto

jQuery

Selectores de formulario.

`$('input')`->Selecciona los elementos `<input>`

`$(':text')` // `$(':checkbox')` // `$(':button')` -> selecciona todos los `<input>` con un atributo como el que se especifica detrás de los `“:”`. En el caso del textarea se incluye dentro de text.

`$(':enabled')`->Selecciona todos los elementos activados.

`$(':disabled')`->Selecciona todos los elementos desactivados.

jQuery

Selectores de formulario

`$(':checked')`->Averigua si los radiobutton y checkbox están chequeados.

`$('objeto').is(':checked')` //devuelve true o false

`$(':radio:checked').attr("value")` //visualiza el valor del radio seleccionado

`$(':radio:checked').val()` //visualiza el valor del radio selec

`$(':selected')`->Selecciona los elementos de la lista de opciones que están seleccionados.

jQuery

Vistos todos los selectores que son un montón vamos a ver algunas funciones que nos pueden ayudar a interactuar con el DOM.

`Val()` —> sin parámetro me devuelve el valor, pero si le introduzco un parámetro le establece ese valor al elemento.

`addClass()` —> agrega la clase pasada por parámetro al elemento.

`removeClass()` —> elimina la clase pasada por parámetro.

`.text()` —> sin parámetro obtiene el contenido de texto del elemento y con parámetro lo establece.

`.html()` —> sin parámetro obtiene el contenido del html y con parámetros lo establece.

`.css("propiedad")` —> solo la propiedad devuelve el valor y dos parámetros dará valor a la propiedad.

`.click()`, `.change()`, `.submit()`, ... —> sirve para manejar eventos.

`.hide()`, `.show()`, —> para ocultar o mostrar el elemento.

`.toggle` —> alterna la visión de un elemento, si está oculto lo muestra y si está visible lo oculta.

`.each(function({}){ })` —> for donde a cada vuelta del bucle se presenta el contenido del elemento.

jQuery

Y por último hablemos del `this`, `$(this)` hará referencia al elemento que con el que estamos trabajando.

```
$('#button').click(function() {  
    console.log('Hiciste clic en:', this);  
});
```

En este caso seleccionamos todos los elementos botón, y cuando hacemos click en uno imprimimos el elemento clicado.

```
$('p').each(function() {  
    console.log('Contenido del párrafo:', $(this).text());  
});
```

En este caso seleccionamos todos los `p`, e iteramos por ellos con un `foreach` que imprime a cada vuelta del bucle el texto del elemento sobre el que se está iterando

jQuery

Bueno vamos a practicar un poco, sobre el html que os proporciono, seleccionar:

- Selecciona todos los párrafos y cambia su texto a "Nuevo texto de párrafo".
- Selecciona el párrafo con el ID "paragraph1" y agrégale la clase "highlight".
- Selecciona todos los elementos de lista () dentro del elemento y cambia su color de texto a verde.
- Selecciona el elemento de entrada de texto con el ID "username" y cambia su valor a "NuevoUsuario".
- Al hacer clic en el botón con el ID "changeTextBtn", cambia el texto de todos los párrafos a "¡Texto cambiado con clic!".
- Selecciona todos los elementos de lista () y ocúltalos.
- Agrega un evento de clic a los párrafos para resaltarlos con un fondo amarillo cuando se hace clic y quitar el resaltado cuando se hace doble clic. (hay una función para doble clickdblclick).
- Selecciona todos los elementos de párrafo con la clase "paragraph" y agrega un borde de 2 píxeles de color rojo.

jQuery

- Selecciona el elemento ancla cuya clase empiece por “seg” y cambia el color del texto a amarillo.
- Cuando se pulse verificar me debe de imprimir por consola si hay algún checkbox están seleccionado.
- Los elementos del dom que estén vacíos ponerles un texto (“texto para elementos vacíos”) en azul.
- Todos los párrafos pares subrayados.
- Los cuatro últimos párrafos con color de fondo verde.
- Selecciona el div que contenga una ul y ponle un borde azul.
- Oculta las descripciones de las imágenes.
- Al hacer click en las imágenes se debe mostrar la descripción si está oculta, si se da click en otra se debe ocultar la descripción del elemento que no se pulso.

jQuery

jQuery tiene filtros que concretan la selección de elementos, por ejemplo:

`not()` -> Para seleccionar aquellos elementos que no cumplen una determinada selección.

```
$('div').not('.verde, #azul').css('border-color', 'red')
```

Selecciona los div que no tengan la clase verde ni el id azul y les cambias el borde a rojo.

jQuery

`filter()` -> Filtrar o buscar una selección de elementos y la reduce a aquellos objetos que sí cumplen con un filtro adicional

```
$("div").filter(".div1").css('background-color', 'blue');
```

Selecciona los `<div>` cuyo ID es `'div1'` y les cambia el color de fondo.

`slice(comienzo, final)`-> Obtiene una porción de una determinada selección de elementos basada en la posición de estos elementos dentro de la selección realizada.

```
$('p').slice(2,3)
```

Primero selecciona todos los elementos `<p>` del documento y luego filtra solo el tercer párrafo.

jQuery

find() -> busca una selección de elementos y la reduce a aquellos objetos que sí cumplen la condición

```
$("#div").find(".div1").css('background-color', 'blue');
```

Selecciona los <div> cuyo ID es 'div1' y les cambia el color de fondo

jQuery

Bucles automáticos

Cuando se aplica una función jQuery a una selección de elementos, no es necesario generar el bucle, puesto que la función lo hace de forma automática.

```
$('.misBotones').hide();
```

En este ejemplo selecciona todos los elementos cuya clase es 'misBotones' y automáticamente se ocultan

jQuery

Encadenar funciones:

jQuery permite utilizar un principio único llamado encadenado, que permite añadir funciones, una después de otra.

```
$('.misBotones').width(300).height(300).text('Hola');
```

En este ejemplo se selecciona todos los elementos cuya clase es 'misBotones' y automáticamente se le aplican tres funciones.

jQuery

jQuery ofrece muchas funciones para manipular elementos y contenido en una página:

`.html()`-> Puede leer el código HTML del interior de un elemento así como sustituir el contenido actual por otro código HTML.

`.text()`-> funciona como `.html()` pero no acepta etiquetas HTML. Es muy útil para reemplazar texto de una etiqueta.

jQuery

Practiquemos un poco lo visto, de un nuevo html que se facilita realiza los siguientes ejercicios:

Selecciona todos los elementos `.item` que no tienen la clase `.special` y ocúltalos.

Selecciona todos los elementos `.special` y cambia su fondo a verde.

Selecciona el segundo elemento `.item` y añade la clase `.highlight`.

Selecciona los elementos `.item` del índice 2 al 4 y cambia su borde a rojo.

Selecciona el contenedor y encuentra todos los elementos `.item` dentro de él y ocúltalos.

jQuery

Podemos también añadir contenido a una página con diferentes métodos:

.append()-> Podremos añadir contenido al último elemento hijo del elemento seleccionado.

Supongamos que disponemos de una lista desordenada que contiene 3 elementos:

```
<ul>  
    <li>Item 1</li>  
    <li>Item 2</li>  
    <li>Item 3</li>  
</ul>
```

Con la siguiente instrucción lo que hacemos es insertar otro elemento al final de la lista desordenada.

```
$('#ul').append('<li>Item 4</li>')
```

jQuery

.after()-> Podremos añadir contenido después del elemento especificado.

Supongamos que tenemos el siguiente código HTML

```
<div class="container">  
  <h2>Saludos</h2>  
  <div class="rojo">Hola</div>  
  <div class="verde">Adiós</div>  
</div>
```

Con la siguiente instrucción lo que hacemos es insertar otro elemento después de la capa rojo

```
$( '.rojo' ).after( '<p>Test</p>' );  
  
<div class="container">  
  <h2>Saludos</h2>  
  <div class="rojo">Hola</div>  
  <p>Test</p>  
  <div class="verde">Adiós</div>  
</div>
```

jQuery

.insertafter()-> Igual que el método after(), la diferencia que primero se especifica el contenido a insertar y después se indica el elemento de referencia para insertar

Supongamos que tenemos el siguiente código HTML

```
<div class="container">
  <h2>Saludos</h2>
  <div class="rojo">Hola</div>
  <div class="verde">Adiós</div>
</div>
```

Con la siguiente instrucción lo que hacemos es insertar otro elemento después de la capa rojo

```
$('<p>Test</p>').insertAfter('.rojo');
```

```
<div class="container">
  <h2>Saludos</h2>
  <div class="rojo">Hola</div>
  <p>Test</p>
  <div class="verde">Adiós</div>
</div>
```

jQuery

.prepend()-> Es igual que .append(), pero añade el código HTML directamente después de la etiqueta de inicio de la selección, es decir, como primer elemento.

```
$('#ul').prepend('<li>Item 0</li>');
```

.before() -> podremos añadir elemento antes del elemento seleccionado.

```
$('#ul').before('<h3>List</h3>');
```

.after()->funciona exactamente igual que la función .before() con la diferencia de que el argumento de la función se añade después del elemento seleccionado

```
$('#ul').after('<h3>before the list</h3>')
```

jQuery

En ocasiones se querrá sustituir o eliminar completamente el elemento seleccionado:

`remove()`

`$('#span.error').remove();` //elimina todas las etiquetas `` de la clase `'error'`.

`empty()`

`$('#div').empty()` //elimina el contenido de todos los `div`

jQuery

`addClass()`-> Añade una clase específica a un elemento.

```
$('div').addClass('misCapas')
```

`removeClass()`-> Elimina una clase especificada del elemento seleccionado.

```
$('div').removeClass ('misCapas')
```

`toggleClass()`-> agrega la clase especificada si no está presente y elimina la clase especificada si está presente.

```
$('div').toggleClass ('misCapas')
```

jQuery

La función `val()` es el equivalente al `value` de las etiquetas que aceptan esta propiedad como los `inputs`, `form`, `textarea...`, es decir, a través de esta función podemos capturar el valor de esos elementos:

```
$(‘selector’).value() //extraer el valor
```

```
$(‘selector’).val() // extraer el valor
```

```
$(‘selector’).val(“Asignar valor”) //asigna valor al elemento
```

jQuery

La funcion `css()` de jQuery permite cambiar directamente las propiedades CSS de un elemento.

```
// establece el color de fondo a una variable  
var bgColor=$('#main').css('background-color')
```

```
// establece una propiedad a un elemento  
$('#body').css('font-size', '200%')
```

```
//Para cambiar varias propiedades a la vez  
$('#miCapa').css({  
    'background-color': '#FF0000',  
    'border': '2px solid #FE0037'  
});
```

jQuery

Para manejar los atributos HTML, jQuery incluye funciones como `attr()` y `removeAttr()`. Ejemplo de atributos - id, class, name, value, href, src, title, style, type ...etc

```
$('.Boton1').attr('value', 'boton1') //asigna el atributo value
```

```
var dato=$('.Boton1').attr('value') //asigna el valor del atributo a una variable
```

```
$('.Boton1').removeAttr('value') // elimina el atributo valu
```

jQuery

La función prop() sirve para modificar propiedades nativas de Javascript de los elementos de una página.

La función removeProp() sirve para eliminar propiedades

Ejemplo de propiedades – readonly, disabled, selected, checked, autofocus ...etc

```
//Añadir o modificar una propiedad  
$('selector').prop('disabled', 'disabled');
```

```
//Eliminar una propiedad  
$('selector').removeProp('disabled');
```

```
//Obtener true o false si tiene esa propiedad  
$('selector').prop('checked');
```

```
//Obtener el índice del elemento seleccionado de un <select>  
$('#Select').prop("selectedIndex" )
```

jQuery

jQuery ofrece la función `.each()` para ejecutar un bucle sobre una selección de elementos.

Si tenemos la siguiente lista ordenada en HTML

```
<ol>  
    <li>Elemento 1</li>  
    <li>Elemento 2</li>  
    <li>Elemento 3</li>  
    <li>Elemento 4</li>  
</ol>
```

Vamos a recorrer la lista y visualizar el contenido:

```
$( 'ol li' ).each( function( ind, elemento ) {  
    console.log( 'El elemento es' + $( "elemento" ).text() + ' el indice es ' + ind );  
})
```

jQuery

Practiquemos, dado un html desarrolla los siguientes ejercicios:

1. Cuando pinche en el botón que pone “Haz click aquí” deberemos seleccionar el checkbox anterior e imprimir por consola el valor de antes de cambiarlo y después del cambio.
2. Elimina el botón que pone elimíname al hacer click en el.
3. Elimina todo el contenido del div que pone “Este div enetero debe desaparecer.”
4. Agrega la clase lista al elemento #anadirClase cuando se haga click sobre el.
5. Añade un parrafo debajo de la lista.
6. Añade otro texto encima del #container4

jQuery

1. Cambia los colores de letra en los párrafos pares a verde y en los impares a gris e imprime por consola los valores de ambos.
2. Inserta un elemento en el contenedor 3 y comprueba en el inspector que se ve antes que el texto.
3. Cambia el elemento “Cámbiame de clase” y ponle una clase al pinchar sobre el y si ya la tiene quitasela.
4. Elimina el Item3 de la lista final.
5. Al pulsar el boton “Cambia el href del ancla anterior”. Cambia el ancla para que dirija a www.ejemplo.com
6. Recorre los elementos de la lista mostrando en consola el contenido y el número de elemento es decir debe de salir en consola
Item 0: Item 1
Item 1: Item 2
El tercer item en principio estaría eliminado sino debe aparecer aquí

jQuery

1. Al pulsar en el botón Elimina href, elimina el atributo href del elemento ancla anterior.
2. Al pulsar el el boton “Cambia el valor del input” se debe cambiar el valor del input anterior y mostrarlo en un console.log.
3. Después del div “InsertarDespues” inserta un párrafo con el texto que quieras.
4. Busca el elemento con la clase highlight y eliminala.

jQuery

Con jQuery podemos también generar efectos. Los efectos JQuery nos permiten darle bonitas animaciones a nuestras páginas que se desencadenan a través de eventos :

hide() y show() -> Permiten ocultar (hide) y mostrar (show)

Sintaxis:

- `$(selector).hide(velocidad, funcion);`
- `$(selector).show(velocidad, funcion);`

Los parametros velocidad y función pueden omitirse.

jQuery

El parámetro velocidad puede tener valores:

‘slow’: esta opción hace que la animación sea más lenta lo que implica una duración más prolongada para mostrar o realizar una transición.

‘fast’: En este caso la animación será más rápida y la duración de la animación más corta.

‘normal’. Representa una velocidad de animación moderada.

También podemos establecer una duración en milisegundos (1000 ms = 1 segundo)

jQuery

Podemos jugar con la función `toggle()` que ya hemos visto para visualizar la animación si estaba oculta o para ocultarla si estaba visible.

La función `toggle` acepta también una velocidad y una función en el caso que lo queramos incluir en los parámetros.

```
$(selector).toggle(velocidad, funcion)
```

jQuery

Tenemos también en esta librería efectos de desplazamiento.

`slideDown()`-> Desplaza el objeto hacia abajo.

`slideUp()`-> Desplaza el objeto hacia arriba.

`slideToggle()`-> Si el elemento está oculto aplica la función `slideDown()` y se va mostrando, si el elemento es visible se aplica la función `slideUp()` y se va ocultado:

```
$(selector).slideDown(velocidad, funcion);
```

```
$(selector).slideUp(velocidad, funcion);
```

```
$(selector).slideToggle(velocidad, funcion)
```

jQuery

Efectos de desvanecimiento:

`fadeIn()`-> Hace que un elemento oculto aparezca mediante un fundido.

`fadeOut()`-> Oculta un elemento mediante un fundido

`fadeToggle()`-> Combina los dos efectos anteriores. Si el elemento está oculto aparece a la vista; si es visible, desaparece.

`fadeTo()`-> Hace que un elemento se disuelva a una opacidad especificada

```
$(selector).fadeIn(velocidad,funcion);  
$(selector).fadeOut(velocidad,funcion);  
$(selector).fadeToggle(velocidad,funcion);  
$(selector).fadeTo(velocidad,opacidad,funcion)
```

jQuery

También encontramos funciones en jQuery para trabajar animaciones:

□ `animate()`-> esta función permite crear muchos tipos de animaciones, tantos como combinaciones de atributos CSS podemos tener

`('selector').animate(Propiedades,[Duración],[FuncióndeAnimacion],[Callback])`

Propiedades: Es obligatorio y es para indicar que atributos de css queremos actualizar.

Duración: Es opcional, se indica la duración de la animación. Pueden ir valores numéricos en milisegundos o "fast" | "slow".

FunciondeAnimacion: Es opcional, esta función es para indicar que tipo de animación se realizará.

Callback: Es opcional, esta función se realiza cuando la animación ha terminado.

jQuery

Practiquemos estos últimos métodos, dado un html realiza los siguientes ejercicios:

1. Los tres primeros botones deben de ocultar mostrar y alternar el elemento de ejemplo, prueba las funciones con diferentes parámetros para la velocidad.
2. Prueba a realizar el mismo ejercicio anterior pero ahora cuando mostramos el elemento en la función que se pasa como parámetro cámbiale el color de fondo a amarillo.
3. En el html hay un div con display a none, muéstralo con la función `slideDown()` y ocúltalo con `slideUp()`, prueba la función `slideToggle` con 5 segundos de retardo y cámbiale el color de fondo al elemento pero dentro de la misma función `slideToggle`.

jQuery

1. Haz uso de los botones Desvanecer para probar las diferentes formas de usar `fadeIn`, `fadeOut`, `fadeToggle` y `fadeTo`.
2. Utiliza el botón de animar para cambiarle el alto y el ancho al div del cuadrado.