

TEMA 5

INTEGRACIÓN DE CONTENIDO INTERACTIVO

Introducción

jQuery es un framework de Javascript. Es decir, librerías de código que contienen procesos o rutinas listos para usar.

Conceptos Básicos de jQuery

`$(document).ready()`

el bloque se ejecutará sólo una vez que la página este disponible.

```
$(document).ready(function() {
```

código

```
});
```

Forma abreviada

(**no recomendada**)

```
$(function() {
```

código

```
});
```


Selección de Elementos en base a su ID

`$('#id');` // **los IDs deben ser únicos**

en base al nombre de clase

`$('.clase');`

`$('#div.clase');` // *si se especifica el tipo de elemento,
se mejora el rendimiento de la selección*

por su atributo

`$('#input[name=first_name]');` // *puede ser muy lento*

en forma de selector CSS

`$('#contents ul.people li');`

Pseudo-selectores

`$('#a.external:first');` // *selecciona el primer elemento <a>*
con la clase 'external'

`$('#tr:odd');` // *selecciona todos los elementos <tr>*
impares de una tabla

`$('#myForm :input');` // *selecciona todos los elementos del tipo input dentro del formulario #myForm*

`$('div:visible');` // *selecciona todos los divs visibles*

`$('div:gt(2)');` // *selecciona todos los divs excepto los tres primeros*

`$('div:animated');` // *selecciona todos los divs actualmente Animados*

`$("a[rel$='thinger']");` // *encontrar todos los <a> cuyo atributo rel terminan en "thinger"*

Comprobar Selecciones

`if ($('#id') { ... }` **no funciona** porque `$()` siempre devuelve un objeto (siempre devuelve true)

Hay que utilizar:

`if ($('#id').length) { ... }`

Guardar Selecciones

`var $divs = $('#div');`

Filtrado de Selecciones

`$('div.foo').has('p');` // el elemento `div.foo` contiene
elementos `<p>`

`$('h1').not('.bar');` // el elemento `h1` no posee la clase `'bar'`

`$('ul li').filter('.current');` // un item de una lista desordenada
que posee la clase `'current'`

`$('ul li').first();` // el primer item de una lista desordenada

`$('ul li').eq(5);` // el sexto item de una lista desordenada

Selección de Elementos de un Formulario

`$('#myForm :input');` // **obtiene todos los elementos inputs #myForm**

:button → Selecciona elementos `<button>` y con el atributo `type='button'`

:checkbox → Selecciona elementos `<input>` con el atributo `type='checkbox'`

:checked → Selecciona elementos `<input>` del tipo checkbox seleccionados

:disabled → Selecciona elementos del formulario que están deshabilitados

:enabled → Selecciona elementos del formulario que están habilitados

:file → Selecciona elementos `<input>` con el atributo `type='file'`

:image → Selecciona elementos `<input>` con el atributo `type='image'`

:input → Selecciona elementos `<input>`, `<textarea>` y `<select>`

:password → Selecciona elementos `<input>` con el atributo `type='password'`

:radio → Selecciona elementos `<input>` con el atributo `type='radio'`

:reset → Selecciona elementos `<input>` con el atributo `type='reset'`

:selected → Selecciona elementos `<options>` que están seleccionados

:submit → Selecciona elementos `<input>` con el atributo `type='submit'`

:text → Selecciona elementos `<input>` con el atributo `type='text'`

Trabajar con Selecciones

Encadenamiento

```
$('#content').find('h3').eq(2).html('texto para el tercer h3');
```

Formateo de código encadenado

```
$('#content')  
    .find('h3')  
    .eq(2)  
    .html('texto para el tercer h3');
```

Restablecer la selección original

```
$('#content')
```

```
.find('h3')
```

```
.eq(2)
```

```
.html('texto para el tercer h3')
```

```
.end() // reestablece la selección a todos los  
        elementos h3 en #content
```

```
.eq(0)
```

```
.html('texto para el primer h3');
```

Getters & Setters (OJO)

Getters

```
$('#h1').html();
```

Setters

```
$('#h1').html('hello world');
```


CSS

Obtener propiedades CSS

```
$('#h1').css('fontSize'); // devuelve "19px"
```

Establecer propiedades CSS

```
$('#h1').css('fontSize', '100px'); // establece una propiedad
```

```
$('#h1').css({  
    'fontSize' : '100px',  
    'color' : 'red'  
}); // establece múltiples propiedades CSS
```

Establecer valores CSS relativos

```
$('#h1').css({
```

```
  'fontSize' : '+=15px', // suma 15px al tamaño  
                           original del elemento
```

```
  'paddingTop' : '+=20px' // suma 20px al  
                           padding superior  
                           original del elemento
```

```
});
```

Trabajar con clases

```
var $h1 = $('h1');
```

```
$h1.addClass('big');
```

```
$h1.removeClass('big');
```

```
$h1.toggleClass('big');
```

```
if ($h1.hasClass('big')) { ... }
```


Dimensiones

`$('h1').width('50px');` // establece el ancho de todos los elementos H1

`$('h1').width();` // obtiene el ancho del primer elemento H1

`$('h1').height('50px');` // establece el alto de todos los elementos H1

`$('h1').height();` // obtiene el alto del primer elemento H1

Atributos

Establecer atributos

```
$('#a').attr('href', 'web.html');
```

```
$('#a').attr({  
    'title' : 'titulo',  
    'href' : 'web.html'  
});
```

Obtener atributos

```
$('#a').attr('href'); // devuelve el atributo href perteneciente al  
primer elemento <a> del documento
```

Recorrer el DOM (varios elementos)

`$('#h1').next('p');` // seleccionar el próximo `<p>` de `h1`

`$('#div:visible').parent();` // seleccionar el a un div visible

`$('#input[name=first_name]').closest('form');` // seleccionar el elemento `<form>` más cercano a un input

`$('#myList').children();` // seleccionar todos los elementos hijos de `#myList`

`$('#li.selected').siblings();` // seleccionar todos los items hermanos del elemento ``

Interactuar en una selección

```
$('#myList li').each(function(idx, el) {  
    console.log('El elemento ' + idx + ' contiene el  
                siguiente HTML: ' + $(el).html());  
});
```

Mover Elementos

`$().insertAfter`, `$().after`, `$().insertBefore`, `$().before`,
`$().appendTo`, `$().append`, `$().prependTo` y
`$().prepend`

// hacer que el primer item de la lista sea el último

`var $li = $('#myList li:first').appendTo('#myList');`

Copiar (clonar) Elementos

`$.fn.clone`

// copiar el primer elemento de la lista y moverlo
al final de la misma

```
$('#myList li:first').clone().appendTo('#myList');
```

Eliminar Elementos

`$.fn.remove`, `$.fn.detach`, `$.fn.empty`

Crear Nuevos Elementos

```
$('#<p>Un nuevo párrafo</p>');
```

```
$('#<li class="new">nuevo item de la lista</li>');
```

Crear un nuevo elemento con atributos utilizando un objeto

```
$('#<a/>', {  
    html : 'Un <strong>nuevo</strong> enlace',  
    'class' : 'new',  
    href : 'web.html'  
});
```

Crear un nuevo elemento en la página

```
var $myNewElement = $('<p>Nuevo elemento</p>');  
$myNewElement.appendTo('#content');
```

Crear y añadir al mismo tiempo un elemento a la página

```
$('ul').append('<li>item de la lista</li>');  
var myItems = [], $myList = $('#myList');  
for (var i=0; i<100; i++) {  
    myItems.push('<li>item ' + i + '</li>');  
}  
$myList.append(myItems.join(""));
```

Manipulación de Atributos

Manipular un simple atributo

```
$('#myDiv a:first').attr('href', 'newDestination.html');
```

Manipular múltiples atributos

```
$('#myDiv a:first').attr({  
    href : 'newDestination.html',  
    rel : 'super-special'  
});
```


El núcleo de jQuery

\$ vs \$()

`$.trim` // **elimina los espacios en blanco del principio y final**

`$('h1').remove();` // **elimina el elemento h1**

El Método Data

Guardar información relacionada a un elemento

`$('#myDiv').data('a', { f : 'b' });` // **guarda en a { foo : 'bar' }**

Recuperar información relacionada a un elemento

`$('#myDiv').data('a');` // **devuelve { f : 'b' }**

Eventos

Vincular un evento utilizando un método reducido

```
$('#p').click(function() {  
    console.log('click');  
});
```

Vincular un evento utilizando el método on

```
$('#p').on('click', function() {  
    console.log('click');  
});
```

Vincular Eventos para Ejecutar una vez (método one)

```
$('p').one('click', function() {  
    console.log('Click al elemento por primera vez');  
    $(this).click(function() {  
        console.log('Se ha clickeado nuevamente');  
    });  
});
```

Desvincular los controladores del evento click

```
$('p').off('click');
```


La función auxiliar hover

```
$('#menu li').hover(function() {  
    $(this).toggleClass('hover');  
});
```

La función auxiliar toggle. Cada vez que un evento ocurre, la función siguiente en la lista se ejecutará

```
$('#p.expander').toggle(  
  function() {  
    $(this).prev().addClass('open');  
  },  
  function() {  
    $(this).prev().removeClass('open');  
  }  
);
```

Efectos

`$().show` → Muestra el elemento seleccionado.

`$('h1').show();`

`$().hide` → Oculta el elemento seleccionado.

`$().fadeIn` → De forma animada, cambia la opacidad del elemento seleccionado al 100%.

`$('h1').fadeIn(300);` // desvanecimiento en 300ms

`$().fadeOut` → De forma animada, cambia la opacidad del elemento seleccionado al 0

`$('h1').fadeOut('slow');`

`$().slideDown` → Muestra el elemento seleccionado con un movimiento de deslizamiento vertical.

`$().slideUp` → Oculta el elemento seleccionado con un movimiento de deslizamiento vertical.

`$('h1').fadeOut('slow');`

`$.slideToggle` → Muestra o oculta el elemento seleccionado con un movimiento de deslizamiento vertical, dependiendo si actualmente el elemento está visible o no.

Ejecutar cierto código cuando una animación haya concluido

```
$('#div.old').fadeOut(300, function() {  
    $(this).remove();  
});
```

Efectos personalizados con \$.animate

```
$('#div.funtimes').animate({  
  left : "+=50",  
  opacity : 0.25  
},  
300, // duration  
function() {  
  console.log('realizado'); // función de devolución de  
  llamada  
});
```


Easing: manera en que un efecto ocurre (si la velocidad durante la animación es constante o no)

Transición de easing por cada propiedad

```
$('#div.funtimes').animate({  
    left : [ "+=50", "swing" ],  
    opacity : [ 0.25, "linear" ]  
},  
300  
);
```

Control de los Efectos

`$().stop` → Detiene las animaciones que se están ejecutando en el elemento seleccionado.

`$().delay` → Espera un tiempo determinado antes de ejecutar la próxima animación.

```
$('h1').show(300).delay(1000).hide(300);
```