

## **Unidad 6. Integración de contenido interactivo**

# Tabla de contenido

- Efectos.....3**
  - Introducción.....3**
  - Efectos Incorporados en la Biblioteca.....3**
    - Cambiar la Duración de los Efectos.....3
    - Realizar una Acción Cuando un Efecto fue Ejecutado.....4
  - Efectos Personalizados con \$.fn.animate.....5**
    - Easing.....5
  - Control de los Efectos.....6**

## EFFECTOS

### Introducción

Con jQuery, agregar efectos a una página es muy fácil. Estos efectos poseen una configuración predeterminada pero también es posible proveerles parámetros personalizados. Además es posible crear animaciones particulares estableciendo valores de propiedades CSS.

Para una completa documentación sobre los diferentes tipos de efectos puede visitar la sección effects: <http://api.jquery.com/category/effects/>.

### Efectos Incorporados en la Biblioteca

Los efectos más utilizados ya vienen incorporados dentro de la biblioteca en forma de métodos: hide y fadeOut añaden un display:none

- **\$.fn.show:** Muestra el elemento seleccionado.
- **\$.fn.hide:** Oculta el elemento seleccionado.
- **\$.fn.fadeIn:** De forma animada, cambia la opacidad del elemento seleccionado al 100%.
- **\$.fn.fadeOut:** De forma animada, cambia la opacidad del elemento seleccionado al 0
- **\$.fn.slideDown:** Muestra el elemento seleccionado con un movimiento de deslizamiento vertical. aparece hacia abajo vertical.
- **\$.fn.slideUp:** Oculta el elemento seleccionado con un movimiento de deslizamiento vertical. desaparece hacia arriba
- **\$.fn.slideToggle:** Muestra u oculta el elemento seleccionado con un movimiento de deslizamiento vertical, dependiendo si actualmente el elemento está visible o no.

cambia de slideDown a slideUp y viceversa

Uso básico de un efecto incorporado  
.fadeTo("slow",0.5) lo deja hasta ese valor de opacidad  
para quitarlo, .fadeTo("slow",1)

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

Cuando se utiliza un efecto de hacer desaparecer un elemento, el elemento no se quita realmente del navegador web. El elemento todavía existe en el DOM o Document Object Model. El código del elemento se encuentra todavía en la memoria del navegador, pero su configuración de pantalla (igual que la configuración de visualización CSS) se establece en none. Debido a que el elemento ya no ocupa ningún espacio visual, el resto del contenido de la página se puede mover a la posición previamente ocupada por el elemento oculto.

### Cambiar la Duración de los Efectos

Con la excepción de \$.fn.show y \$.fn.hide, todos los métodos tienen una duración predeterminada de la animación en 400ms. Este valor es posible cambiarlo.

## Configurar la duración de un efecto

```
$( 'h1' ).fadeIn(300); // desvanecimiento en 300ms
$( 'h1' ).fadeOut('slow'); // utilizar una definición de velocidad interna
```

### *jQuery.fx.speeds*

jQuery posee un objeto en `jQuery.fx.speeds` el cual contiene la **velocidad predeterminada para la duración de un efecto**, así como también los valores para las **definiciones “slow” y “fast”**.

```
speeds: {
  slow: 600,
  fast: 200,
  // velocidad predeterminada
  default: 400
}
```

Por lo tanto, es posible sobrescribir o añadir nuevos valores al objeto. Por ejemplo, puede que quiera cambiar el valor predeterminado del efecto o añadir una velocidad personalizada.

### **Añadir velocidades personalizadas a `jQuery.fx.speeds`**

```
jQuery.fx.speeds.muyRapido = 100;
jQuery.fx.speeds.muyLento = 2000;
```

fx espacio de nombres, para agrupar funciones, speeds esta dentro

## Realizar una Acción Cuando un Efecto fue Ejecutado

A menudo, querrá ejecutar una acción una vez que la animación haya terminado — ya que si ejecuta la acción antes que la animación haya acabado, puede llegar a alterar la calidad del efecto o afectar a los elementos que forman parte de la misma. [Definición: Las funciones de devolución de llamada (en inglés **callback functions**) proveen una forma para **ejecutar código una vez que un evento haya terminado**.] En este caso, el evento que responderá a la función será la conclusión de la animación. Dentro de la función de devolución, la palabra clave `this` hace referencia al elemento en donde el efecto fue ejecutado y al igual que sucede con los eventos, es posible transformarlo a un objeto jQuery utilizando `$(this)`.

Ejecutar cierto código cuando una animación haya concluido

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

**Note que si la selección no retorna ningún elemento, la función nunca se ejecutará.** Este problema lo puede resolver **comprobando si la selección devuelve algún elemento**; y en caso que no lo haga, ejecutar la función de devolución inmediatamente.

### Ejecutar una función de devolución incluso si no hay elementos para animar

```
var $thing = $('#nonexistent');

var cb = function() {
    console.log('realizado');
};

if ($thing.length) {
    $thing.fadeIn(300, cb);
} else {
    cb();
}
```

### Efectos Personalizados con \$.fn.animate

.animate( por cada propiedad [, duration] [, easing] [, complete] )

Es posible realizar animaciones en propiedades CSS utilizando el método \$.fn.animate. Dicho método permite realizar una animación estableciendo valores a propiedades CSS o cambiando sus valores actuales.

Efectos personalizados con \$.fn.animate

```
$('#div.funtimes').animate(
    {
        left : "+=50",    las propiedades que se pueden animar aqui son aquellas que tienen numeros
        opacity : 0.25
    },
    300, // duration
    function() { console.log('realizado'); // función de
    // devolución de llamada
});
```

### Nota

Las propiedades relacionadas al color no pueden ser animadas utilizando el método \$.fn.animate, pero es posible hacerlo a través de la extensión color plugin que se puede lograr incluyendo la librería jQueryUI en la página web.

**Easing** cargar la librería jQueryUI para personalizar los easing

El concepto de Easing describe la manera en que un efecto ocurre, es decir, si la velocidad durante la animación es constante o no. jQuery incluye solamente dos métodos de easing: **swing** y **linear**. Si desea transiciones más naturales en las animaciones, existen varias extensiones que lo permiten.

A partir de la versión 1.4 de la biblioteca, es posible establecer el tipo de transición por cada propiedad utilizando el método \$.fn.animate.

### Transición de easing por cada propiedad

```
$('#div.funtimes').animate(
    {
        left : [ "+=50", "swing" ],
```

```
        opacity : [ 0.25, "linear" ]
    },
    300
);
```

Para más detalles sobre las opciones de easing, consulte <http://api.jquery.com/animate/> y <http://jqueryui.com/easing/> donde se encontrarán las funciones de easing del api de jQueryUI.

## Control de los Efectos

jQuery provee varias herramientas para el manejo de animaciones.

- **\$.fn.stop:** Detiene las animaciones que se están ejecutando en el elemento seleccionado.
- **\$.fn.delay:** Espera un tiempo determinado antes de ejecutar la próxima animación.

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

- **jQuery.fx.off:** Si el valor es verdadero (true), no existirán transiciones para las animaciones; y a los elementos se le establecerá el estado final de la animación. Este método puede ser especialmente útil cuando se está trabajando con navegadores antiguos.

.stop(true, false) el true indica que se paren todas las animaciones, y el false indique que se quede tal cual, en el estado en el que esta  
.stop(true,true) parar todas las animaciones pero se quedan en su estado final, terminan