

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Efectos y Animaciones © ADRINFOR S.L.

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Indice

| | |
|-------------------------------------|-----------|
| Efectos y Animaciones | 3 |
| Introducción | 3 |
| Cuándo utilizar animaciones | 3 |
| Opciones comunes de las animaciones | 4 |
| Mostrar y ocultar elementos | 5 |
| Método .show() | 6 |
| Método .hide() | 7 |
| Método .toggle() | 8 |
| Modificando la opacidad | 9 |
| Métodos .fadeIn() y .fadeOut() | 10 |
| Método .fadeTo() | 10 |
| Método .fadeToggle() | 11 |
| Creando animaciones personalizadas | 12 |
| Parámetros del método .animate() | 12 |
| Ejercicios | 15 |
| Ejercicio 1: Pestañas | 15 |
| Lo necesario para comenzar | 15 |
| Ejercicio 2: Memorion | 16 |
| Lo necesario para comenzar | 16 |

Efectos y Animaciones

Introducción

jQuery nos ofrece una amplia gama de métodos con los que realizar animaciones, completamente funcionales y que no necesitan la inclusión de extensiones adicionales para funcionar. Estos métodos controlan animaciones de deslizamiento y "decoloración", fácilmente utilizables dado que requieren una configuración de funcionamiento mínima, y que será soportada por la mayoría de los navegadores actuales.

También podremos encontrar métodos con los que controlar la cola de animaciones de la página, así como la posibilidad de crear animaciones personalizadas que se apoyarán sobre estilos numéricos de CSS.



Cuándo utilizar animaciones

Las animaciones pueden conseguir una gran mejora en la experiencia que un usuario obtiene en una web. Sin embargo, hay ocasiones en las que sobrecargar la página de animaciones puede ser contraproducente o poco recomendado.

Esta situación la encontramos, por ejemplo, en acciones que los usuarios de la web repetirán en múltiples ocasiones y en las que la incorporación de una animación puede resultar molesta. Por ello, el uso de animaciones puede estar correctamente justificado en las siguientes situaciones:

1

Al mostrar u ocultar ventanas emergentes y paneles de contenido

2

Cuando un elemento es movido a una posición diferente de la página

3

Al cambiar algún estado en la página debido a una interacción con el usuario

4

Cuando algún elemento está siendo transformado o cambiando su estado

5

Para llamar la atención del usuario sobre un elemento específico de la página o resaltar algún elemento con el que deba interactuar.

Antes de implementar una animación en tu página, realiza las siguientes preguntas para decidir si utilizar o no una animación en un contexto:

- ¿es una animación práctica?
- ¿la duración de la animación es adecuada?
- ¿añade la animación valor a la experiencia de usuario en el sitio?

Si la respuesta a las anteriores preguntas es positiva entonces podemos considerar que sí es una buena opción implementar una animación. Si por el contrario alguna de las preguntas anteriores es respondida negativamente entonces debes replantearte la necesidad de incluir una animación.



Ten en cuenta que estas anotaciones son únicamente recomendaciones. Ante todo intenta usar el sentido común, y que la página no pierda su funcionalidad únicamente por conseguir vistosidad con animaciones innecesarias.

Opciones comunes de las animaciones

Los métodos de animación predefinidos en jQuery pueden ser llamados indicando como parámetro un objeto anónimo en el que, como nombre de las propiedades indicadas en el mismo, se podrán utilizar las siguientes:

duration

Número que determinará la duración de la animación. El valor por defecto de este parámetro será 400 (0.4s)

easing

Parámetro que indicará el método de aceleración a utilizar para la animación. jQuery incluye en su composición básica dos métodos: swing y linear. La opción swing será la utilizada por defecto por la librería

queue

Este parámetro indicará si se deberá o no añadir la animación a la cola de efectos. Si se indica el valor false, la animación comenzará inmediatamente.

Otra opción es indicar el valor true, o una cadena si queremos especificar una cola de efectos concreta o utilizar la cola de efectos predefinida. En este caso, la animación esperará a su turno para ejecutarse.



La opción de definir la cola de efectos específica se incorporó en la versión 1.7 de jQuery. Si utilizas una versión anterior de la librería, esta opción no estará disponible.

step

Función que será invocada para cada propiedad animada en cada elemento animado.

progress

Esta opción, disponible a partir de la versión 1.8 de jQuery proporcionará la posibilidad de ejecutar una función para cada paso de la animación, una única vez por cada elemento animado independientemente del número de propiedades animadas.

complete

Función que se ejecutará una vez la animación se haya completado.

start

Función que será ejecutada cuando la animación comience. Esta opción fue añadida en la versión 1.8 de jQuery.

done

Función que se ejecutará cuando la animación se complete.

Disponible desde jQuery 1.8, recibirá dos parámetros: Promise animation, Boolean jumpedToEnd

fail

Función que será ejecutada cuando ocurra algún error en la animación.

Al igual que las opciones anteriores, esta opción fue añadida en la versión 1.8 de jQuery.

always

Función que se llamará cuando la animación finalice, independientemente de que haya finalizado correctamente o no se haya realizado por completo.

Disponible desde jQuery 1.8

Mostrar y ocultar elementos

Uno de los tipos de animación más básicos, y a la vez más útiles que incluye jQuery son aquellas animaciones que en sus llamadas se encargan de mostrar y ocultar elementos. Estas animaciones se consiguen por medio de tres métodos:

- Método .show()
- Método .hide()
- Método .toggle()

Por defecto, las llamadas a los métodos .show() y .hide() modificarán la visibilidad de los elementos sobre los que se apliquen modificando la propiedad CSS display de los mismos. Al llamar a estos métodos sin argumentos adicionales la modificación será instantánea y no producirá una animación visual a parte de lo indicado anteriormente.

```
$('#btnMostrar').click(function(){
    $('.elementosOcultos').show();
});
```

```
$('#btnOcultar').click(function(){
    $('.elementosVisibles').hide();
});
```

Veamos todas las opciones disponibles en cada uno de los métodos para poder configurar su funcionamiento.

Método .show()

Con el método .show() conseguiremos que los elementos pertenecientes a la selección sobre la que se aplique sean mostrados.

\$(selector).show()

Como hemos indicado anteriormente, una forma de llamar a este método será sin indicarle ningún parámetro. De este modo, todos los elementos incluidos en la selección \$(selector) serán mostrados inmediatamente sin realizar ninguna animación en el proceso. Internamente, jQuery recuperará el valor de la propiedad CSS "display" de cada uno de los elementos, estableciendo su valor original o el valor "block" si la propiedad no hubiese estado asignada.

\$(selector).show(duracion)

Siempre que al método .show() le proporcionemos algún parámetro, la acción de ocultar se realizará por medio de una animación. Esta animación modificará las propiedades width, height y opacity de forma simultánea de los elementos sobre los que se aplique.

Una forma de invocar al método controlando la duración de la animación será indicándole, como primer parámetro, la duración de la animación. Este valor será indicado en milisegundos:

| |
|--|
| Indicando el valor 100, la duración será de 0,1 segundos |
| indicando el valor 1000 la duración será de 1 segundos |
| etc. |

\$(selector).show(function() {/**/})

Otro parámetro que podrá recibir el método será una función controladora, que capturará el momento en el que la animación se complete, momento en el que se ejecutará el código de la misma.



En este caso, la animación utilizará el valor por defecto para la duración de la misma cuyo valor es de 400ms (0.4s).

`$(selector).show(duracion, function(){//})`**

Estos dos parámetros podrán combinarse para conseguir una llamada al método más personalizada. De este modo, controlaremos tanto la duración de la animación como duracion, además de poder ejecutar una función cuando ésta finalice.

`$(selector).show(duracion, aceleracion, function(){//})`**

Adicionalmente a la configuración anterior, el método `.show()` podrá recibir como segundo parámetro una cadena indicando el tipo de aceleración a aplicar en la animación. En este caso, la función controladora del fin de la animación será relegada a la tercera posición.

`$(selector).show(options)`

Por último, esta forma de llamar al método permitirá su configuración máxima. En esta ocasión, el parámetro indicado al método será un objeto anónimo, que configurará la animación a partir de las opciones comunes disponibles para los métodos de animación.

Método `.hide()`

El método `.hide()` ocultará los elementos incluidos como resultados en el objeto jQuery sobre el que sea aplicado.

`$(selector).hide()`

Al llamar al método sin parámetros, los elementos existentes en la selección `$(selector)` se ocultarán inmediatamente sin aplicar una animación en el proceso. Este método aplicará a la propiedad CSS "display" el valor "none", almacenando como caché de datos el elemento su valor anterior para, en caso de tener que volver a mostrar los elementos, poder recuperar su valor original.

`$(selector).hide(duracion)`

Cuando el método reciba algún parámetro, el proceso se realizará mediante una animación que modificará las propiedades width, height y opacity de los elementos existentes en la selección hasta que éstos dejen de estar visibles en la página.

Para definir la duración de ésta animación podremos llamar al método con un parámetro, que indicará la duración de la animación. Este valor se indicará en milisegundos:

| |
|--|
| Indicando el valor 100, la duración será de 0,1 segundos |
| indicando el valor 1000 la duración será de 1 segundos |
| etc. |

También podremos utilizar las cadenas predefinidas "normal", "fast" y "slow". Cada una de ellas definirá una duración diferente como se muestra en la siguiente tabla:

| Cadena | Valor de la animación |
|----------|-----------------------|
| "normal" | 400 ms |
| "fast" | 200 ms |
| "slow" | 600 ms |

Cuanto mayor sea el valor de la animación, el efecto se realizará más despacio.

`$(selector).hide(function(){//})`**

Otra forma de llamar al método `.hide()` será indicándole como parámetro una función controladora que se encargará de capturar el evento de completado de la animación y ejecutará la misma cuando éste ocurra.



Al no indicar un valor de duración, la animación se realizará utilizando la duración por defecto de 400ms (0.4s).

`$(selector).hide(duracion, function(){//})`**

Estos dos parámetros podrán combinarse para conseguir una llamada al método más personalizada. De este modo, controlaremos tanto la duración de la animación como `duracion`, además de poder ejecutar una función cuando ésta finalice.

`$(selector).hide(duracion, aceleracion, function(){//})`**

Adicionalmente a la configuración anterior, el método `.show()` podrá recibir como segundo parámetro una cadena indicando el tipo de aceleración a aplicar en la animación. En este caso, la función controladora del fin de la animación será relegada a la tercera posición.

`$(selector).hide(options)`

Por último, esta forma de llamar al método permitirá su configuración máxima. En esta ocasión, el parámetro indicado al método será un objeto anónimo, que configurará la animación a partir de las siguientes propiedades:



Pese a que jQuery solamente ofrece dos posibles valores de aceleración según su implementación básica, es posible ampliar los tipos de aceleración por medio de plugins. Puedes encontrar una gama de efectos muy interesante en la página del proyecto [jQuery UI suite](#).

Método `.toggle()`

El método `.toggle()` combinará las funcionalidades de los métodos `.show()` y `.hide()`. Al aplicar el método sobre una selección, los elementos incluidos en la misma modificarán su visibilidad:

- Si estuviesen ocultos, serán mostrados
- Si hubiesen estado visibles, se ocultarán



Este método también se encuentra disponible como controlador de un método. Dependiendo de los argumentos que reciba el método, jQuery proveerá una u otra funcionalidad.

La morfología de este método es muy similar a la disponible en los métodos `.show()` y `.hide()`. Soporta los mismos parámetros que estos dos:

- **`$(selector).toggle()`** - sin realizar ningún tipo de animación en la transición, los elementos existentes en la selección que reciban la acción cambiarán su estado de visibilidad.
- **`$(selector).toggle(duracion)`** - Al indicarle un parámetro numérico, éste se considerará como el tiempo en milisegundos que deberá durar la animación en el proceso de cambio de estado de cada elemento. Este valor se indicará en milisegundos y seguirá las mismas reglas que en los métodos anteriormente vistos.
- **`$(selector).toggle(duracion, function() {/**/})`** - Al indicarle una función como segundo parámetro, ésta se considerará como controladora de completado de la animación. Al finalizar la animación ésta será ejecutada.
- **`$(selector).toggle(duracion, aceleracion, function() {/**/})`** - Si como segundo parámetro se le indica una cadena, ésta describirá el tipo de aceleración a aplicar en la animación. En este caso, la función controladora del fin de la animación será relegada a la tercera posición.
- **`$(selector).toggle(options)`** - Esta forma de llamar al método permitirá su configuración más extensa. El parámetro será un objeto anónimo que podrá utilizar los mismos parámetros de configuración que los métodos `.show()` y `.hide()`.

Además de todos estos modos de llamar al método, éste aceptará uno adicional no disponible en los métodos anteriormente vistos.

`$(selector).toggle(booleano)`

El método `.toggle()` también podrá ser llamado indicándole como parámetro un valor booleano (`true` o `false`). Cuando se le indique el valor `true`, los elementos de la selección serán mostrados y, en el caso de indicar el valor `false`, éstos se ocultarán.

Modificando la opacidad

Otro tipo de animación que podremos realizar con jQuery modificará la opacidad de los elementos animados. Este tipo de animación conseguirá visibilidad nuestras animaciones, consiguiendo un efecto sobrio y elegante. Para incorporar de forma sencilla este efecto en nuestras páginas, jQuery ofrece cuatro métodos que, sin necesidad de realizar configuraciones complejas, conseguirán el efecto de modificación de opacidad en nuestro sitio. Los métodos son:

- Métodos `.fadeIn()` y `.fadeOut()`
- Método `.fadeTo()`
- Método `.fadeToggle()`

Métodos `.fadeIn()` y `.fadeOut()`

Los métodos `.fadeIn()` y `.fadeOut()` ofrecerán un método sencillo con el que poder modificar completamente la opacidad de elementos del DOM. Por un lado, el método `.fadeIn()` animará los elementos incrementando su opacidad hasta el 100%, mientras que el método `.fadeOut()` reducirá la opacidad hasta llegar al 0% y conseguir que los elementos dejen de ser visibles.

Ambos métodos podrán ser llamados con los mismos tipos de argumentos, consiguiendo los mismos resultados (con la única diferencia de que el método `.fadeIn()` mostrará y el método `.fadeOut()` ocultará) según los parámetros que reciban.

`$(selector).fadeIn(duracion, function(){//})` y `$(selector).fadeOut(duracion, function(){/**/})`**

La primera forma de llamar al método es con un máximo de dos parámetros:

- El primer parámetro indicará a ambos métodos la duración en milisegundos que debe tener la animación. Este parámetro opcional tomará por defecto el valor 400, haciendo que la animación dure 0.4 segundos.
- El segundo parámetro, también opcional, permitirá definir una función que se ejecutará cuando la animación se complete.

`$(selector).fadeIn(duracion, aceleracion, function(){//})` y `$(selector).fadeOut(duracion, aceleracion, function(){/**/})`**

Otra forma de llamar al método será indicándole un máximo de tres parámetros, todos ellos opcionales, con el siguiente orden:

- El primer parámetro indicará la duración en milisegundos de la animación. Por defecto utilizará el valor 400, que será utilizado si no es indicado.
- En segundo lugar indicaremos un método de aceleración, pudiendo aceptar dos tipos de aceleración: "linear" o "swing".
- Por último, podremos indicar una función de completado que será ejecutada al completarse el efecto.

`$(selector).fadeIn(options)` y `$(selector).fadeOut(options)`

Por último, esta configuración de los métodos permitirá su personalización máxima. El parámetro será un objeto anónimo que podrá utilizar las opciones comunes de las animaciones. Estas opciones se le indicarán al método en forma de objeto anónimo, en el que los atributos indicarán la opción, y su valor, el valor de la opción.

Método `.fadeTo()`

El método `.fadeTo()` aplicará opacidad a los elementos seleccionados en los que se aplique. Para que la animación sea realizada, el método recibirá, como mínimo, dos parámetros:

- La duración de la animación, indicada en milisegundos

- La opacidad hasta la que llegará la animación

Con estos dos parámetros definiremos el comportamiento básico de la animación. Pese a ello, el método podrá aceptar hasta un máximo de cuatro parámetros (dos más de los anteriormente indicados), que permitirán que el método sea invocado de tres modos distintos según los tipos de datos indicado como parámetros opcionales:

.fadeTo(duracion, opacidad)

Como hemos comentado anteriormente, los dos primeros parámetros del método son obligatorios y necesarios para que el método funcione correctamente. Con estos dos parámetros el método aplicará la animación, modificando la opacidad de los elementos seleccionados hasta alcanzar el valor indicado en el segundo parámetro, y tardará el tiempo indicado como duración.

.fadeTo(duracion, opacidad, complete)

Cuando al método se le indique como tercer parámetro una función ésta actuará como controladora del completado de la animación, siendo ejecutada cuando finalice.

.fadeTo(duracion, opacidad, aceleracion, complete)

También se podrá llamar al método indicando como tercer parámetro el tipo de aceleración que se debe aplicar a la animación. En caso de indicar este tercer parámetro también podremos indicarle al método una función anónima como cuarto parámetro, que al igual que en el tipo de llamada anterior, se ejecutará al completarse la animación.

Método .fadeToggle()

Al igual que ocurría con el método .toggle() para los métodos encargados de mostrar y ocultar elementos, el método .fadeToggle() ofrecerá una combinación de los métodos .fadeIn() y .fadeOut(), realizando una u otra acción dependiendo del estado de visibilidad de cada uno de los elementos sobre los que se aplique:

- Si un elemento se encuentra oculto, éste será mostrado por medio de una animación que aumente su opacidad.
- Si un elemento se encuentra visible, la animación lo ocultará, reduciendo durante la animación la opacidad del mismo.

El método .fadeToggle() podrá ser llamado de dos modos:

.fadeToggle([duration] [, easing] [, complete])

Este tipo de llamada aceptará un máximo de tres parámetros, en el siguiente orden:

1. Duración de la animación
2. Efecto de aceleración de la animación
3. Función controladora de completado de la animación

.fadeToggle(options)

La otra forma de llamar al método es indicándole como parámetro un objeto anónimo, en el que los nombres de sus atributos se correspondan con alguna de las opciones comunes de animaciones vistas anteriormente.

Al llamar al método de este modo podremos indicar detalladamente las opciones con las que deseemos controlar la animación.

Creando animaciones personalizadas

Además de las animaciones vistas hasta ahora, jQuery provee métodos con los que poder hacer animaciones personalizadas con los elementos del DOM. Esta funcionalidad es ofrecida por el método `.animate()`, que conseguirá sorprendentes efectos por medio de la modificación de las propiedades CSS de los elementos a animar.

El método `.animate()` podrá ser utilizado para conseguir efectos animados por medio de la modificación de propiedades CSS numéricas.



Aunque el modo de uso del método puede tener similitudes con el método `.css()` el resultado obtenido es completamente distinto, ya que este método únicamente establecerá las propiedades directamente, mientras que el método `.animate()` realizará la asignación de nuevos valores mediante una animación.

Este método es una opción polivalente ya que podremos crear diversos tipos de animaciones, realizando combinaciones de los atributos CSS a modificar, y jQuery se encargará de modificar los atributos indicados por medio de una transición.

Parámetros del método `.animate()`

Al hacer una llamada al método `.animate()` deberemos indicarle una serie de parámetros con los que definir el funcionamiento de la animación.

`.animate(Propiedades, [Duración], [Función de animación], [Callback])`

Propiedades

El primer parámetro, de carácter obligatorio, indicará al método qué atributos CSS queremos actualizar, así como los nuevos valores a establecer en cada uno de ellos. Este parámetro deberá ser de tipo objeto anónimo, y entre los atributos CSS que puede contener solamente serán aceptados aquellos que tengan valores numéricos.

Los atributos a establecer pueden ser indicados con un valor absoluto o como una operación aritmética contenida en una cadena de texto, como `"+=50"` o `"-=50"`, valores con los que, el valor actual de la propiedad, incrementará o se reducirá en 50.



Los modos resumidos de CSS no son completamente soportados por el método. Es recomendable indicar los atributos a modificar en su forma extendida.

Duración

El segundo parámetro del método permitirá indicar la duración de la animación. Cuanto mayor sea el valor asignado en este parámetro, la animación será más lenta y, por el contrario, cuanto inferior sea el valor, más rápidamente se realizará la animación.

Podremos indicar este parámetro de dos modos:

- Como un número, que indicará la duración en milisegundos de la animación
- Con una de las etiquetas "normal", "fast" o "slow", para utilizar estos valores predefinidos según la siguiente tabla

| Cadena | Valor de la animación |
|----------|-----------------------|
| "normal" | 400 ms |
| "fast" | 200 ms |
| "slow" | 600 ms |

En caso de no indicar este parámetro, la animación se realizará con la duración por defecto siendo ésta de 400 milisegundos (0.4s).

Función de animación

El tercer parámetro indicará al método el tipo de aceleración de la animación. Los posibles valores de este parámetro serán:

- swing, para que la animación sea más suavizada al comienzo y tenga un final más rápido
- linear, para que el efecto de animación sea constante.

En caso de no indicar ningún valor para este parámetro, la aceleración tomará el valor "swing" como valor por defecto.

Función de Completado

El último parámetro que podremos indicar al método nos ofrecerá la posibilidad de ejecutar una función cuando se haya terminado totalmente de producir el efecto. En otras palabras, esta ejecución será invocada cuando los atributos CSS indicados en el primer parámetro alcancen los valores indicados. Si la animación afecta a varios elementos, ésta función será ejecutada una vez por cada elemento que sea animado.

El uso de este parámetro puede ser una opción muy útil si queremos encadenar animaciones en secuencia.



A diferencia de los métodos de animación predefinidos - como `.slideDown()` y `.fadeIn()` - el método `.animate()` no modificará la visibilidad de los elementos. Una animación aplicada sobre un elemento oculto modificará las propiedades del mismo, pero se mantendrá oculto a pesar de ésta.

Ejercicios

Ejercicio 1: Pestañas

Duración estimada del ejercicio



20
minutos

Añade el código jQuery necesario para dotar al HTML de la página paginacontabs.htm de funcionalidad de pestañas.

La página contiene un elemento `ul.tabs` con tres encabezados, así como un `div.tabsContainer` que contiene el mismo número de elementos que el `ul.tabs`.

```
<div id="wrapper">
  <ul class="tabs">
    <li><a href="#" class="selected">Tab #1</a></li>
    <li><a href="#">Tab #2</a></li>
    <li><a href="#">Tab #3</a></li>
  </ul>
  <div class="tabsContainer">
    <div class="tab-content">Contenido del primer tab</div>
    <div class="tab-content">Contenido del segundo tab</div>
    <div class="tab-content">Contenido del tercer tab</div>
  </div>
</div>
```

Una vez incluido el código en la página, al pulsar los elementos del encabezado se deberá mostrar el elemento descendiente del `div.tabsContainer` con mismo índice. Haz uso de los métodos `.show()` y `.hide()` para realizar las animaciones de mostrar / ocultar los elementos.



Para conocer el elemento a mostrar, haz uso de los métodos `.index()` y `.eq()` de jQuery.

Lo necesario para comenzar

Utiliza el archivo `ud10ej01.htm` para incluir el código jQuery que de funcionalidad a la página.



ingredientes-ejercicio-pestanas.zip

Extrae el contenido del archivo
en tu directorio de trabajo.

Ejercicio 2: Memorion

Duración estimada del ejercicio



Este ejercicio propone un reto complejo de uso de jQuery. En el documento memorion.htm disponemos de un fragmento de código JavaScript, que define un Array de imágenes que deberán ser utilizadas para componer un cuadrado con el que formar un juego de memoria, en el que las imágenes estarán ocultas y, al ser pulsado el elemento que las contengan, ser mostradas.

Se deberán seleccionar imágenes de dos en dos, y en caso de contener la misma imagen, éstas permanecerán visibles. En caso contrario, las imágenes deberán de volver a ocultarse.

Además, deberemos contabilizar el número de intentos, asignando éstos dentro del span#counter

Lo necesario para comenzar

Descarga el archivo memorion.zip y extrae el contenido en el directorio de trabajo. Este archivo contendrá una lista con las imágenes, que deberán ser cargadas dentro del div#boxcard



memorion.zip

Plantilla para el juego Memorion