JavaScript

ANEXO 6.-

JQuery:

Efectos

DISTRIBUIDO POR:

CENTRO DE INFORMÁTICA PROFESIONAL S.L.

C/ URGELL, 100
08011 BARCELONA
TFNO: 93 426 50 87

C/ RAFAELA YBARRA, 10
48014 BILBAO
TFNO: 94 448 31 33

www.cipsa.net

Manejo de Efectos

JQuery permite fácilmente añadir diferentes efectos a las páginas web, e incluso crear efectos personalizados animando propiedades CSS.

Transiciones

JQuery dispone de funciones que permiten generar transiciones haciendo que los elementos sobre las que se aplican aparezcan o desaparezcan de manera gradual o en un intervalo de tiempo.

Las siguientes funciones pueden invocarse de tres modos:

- Sin ningún argumento → Provoca la aparición y ocultación inmediata.
- Con argumentos "***slow***", "***normal***", "***fast***" → Provoca la aparición y ocultación del elemento mediante una transición lenta, normal, o rápida respectivamente.
- Con argumento numérico → Equivalente al anterior, pero el efecto dura los milisegundos indicados como argumento.

Efecto de desvanecimiento (*$().fadeIn(), $().fadeOut()*)

Estas funciones ocultan o muestran los elementos de la selección contra la que se invocan *mediante una transición de opaco a transparente, o viceversa*. La velocidad de la transición viene depende del argumento dado tal como en ***$().show()*** y ***$().hide()***.

```
// Oculto todos los párrafos en 1.5 segundos con una transición transparente.
$("p").fadeOut(1500);
// Muestra todas las capas ocultas con una transición de 0.75 segundos.
$("div:hidden").fadeIn(750);
```

Efecto de escalado (*$().slideUp(), $().slideDown()*)

Estas funciones ocultan y muestran los elementos de la selección mediante un escalado de la anchura y la altura. La velocidad del escalado depende del argumento indicado.

```
// Escala todos los párrafos al 0% en 0.8 segundos.
$("p").slideUp(800);
// Muestra todas las capas ocultas escalándolas a su tamaño original en 0.6 segs-
$("div:hidden").slideDown(600);
```

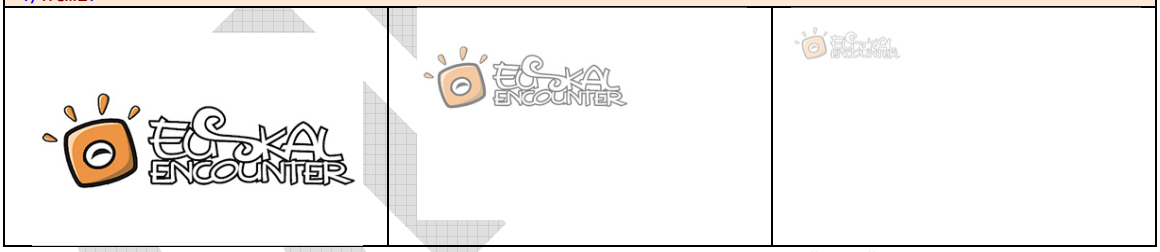
(*) Para poder seleccionar los elementos ocultos y los visibles pueden emplearse los selectores "*:hidden*" y "*:visible*" respectivamente.

Efecto de transición (*$().show(), $().hide()*)

JQuery dispone de las las funciones *$().show()* / *$().hide()* para mostrar y ocultar respectivamente los elementos de la selección contra la que se invoca. Ambos métodos pueden invocarse de tres maneras:

Ejemplo: El siguiente código muestra una imagen y dos botones "*Ocultar*" y "*Mostrar*" que al ser pulsados provocan la ocultación y muestra de la imagen:

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title></title>
    <script src="Scripts/jquery-3.0.0.js"></script>
    <script>
        $().ready(function () {
            $("#btnOculto").click(function () {
                // Ocultacion lenta.
                $("img").hide("slow");
            });
            $("#btnVisible").click(function () {
                // Mostrado rápido
                $("img").show("slow");
            });
        });
    </script>
</head>
<body>
    <img src="images/imagen0.png" /><br />
    <input type="button" id="btnOculto" value="OCULTO" />
    <input type="button" id="btnVisible" value="VISIBLE" />
</body>
</html>
```



Animacion de desvanecimiento de elemento con funcion S().hide()

Cambio de estado (*$().toggle(), $().fadeToggle(), $().slideToggle()*).

Estas funciones ocultas y muestran los elementos en función de su estado, es decir; si son visibles los ocultan, y si están ocultos los muestran. El efecto aplicado depende de la función empleada:

- ***$().toggle()*** → Emplea efecto de desvanecimiento y escalado como *$().show* y *$().hide*.
- ***$().fadeToggle()*** → Emplea efecto de desvanecimiento como *$().fadeOut()* y *$().fadeIn()*
- ***$().slideToggle()*** → Emplea efecto de escalado como *$().slideDown()* y *$().slideUp()*

Ejemplo: El siguiente código muestra una imagen que se desvanece y reaparece mediante una transición de 1 segundo cada vez que se pulsa el botón "VER/OCULTAR".

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title></title>
    <script src="Scripts/jquery-3.0.0.js"></script>
    <meta charset="utf-8" />
    <script>
        $().ready(function () {
            $("#btnAccion").click(function () {
                // Si la imagen está oculta, la muestra. Si es visible, la oculta.
                $("img").fadeToggle(1000);
            });
        });
    </script>
</head>
<body>
    <img src="images/imagen0.png" /><br />
    <input type="button" id="btnAccion" value="VER/OCULTAR" />
</body>
</html>
```

Sincronización de acciones.

Existen ocasiones en las que es interesante que se ejecute una determinada operación una vez que un efecto se complete. Para ello podríamos intentar lo siguiente:

```
$("#btnOculto").click(function () {
    // Ocultacion lenta.
    $("img").hide("slow");
    // Visualizacion del mensaje.
    alert("OCULTO");
});
```

Sin embargo, con este código lo que sucederá es que la sentencia *alert()* se ejecuta inmediatamente tras la iniciarse la animación de ocultación del elemento, y no al final. Veremos por tanto el mensaje "OCULTO" antes de que realmente se haya ocultado el elemento *<img>*.

Para sincronizar una acción al momento en que termina un efecto puede indicarse una *función de retrollamada* como segundo argumento de la función efecto:

Ejemplo: El siguiente código muestra un efecto de ocultación lenta con una función de retrollamada asociada que muestra el mensaje "TERMINADO" una vez oculto el elemento. Para ello se invoca la función *$().hide()* indicando como segundo argumento una función que se ejecutará estrictamente una vez se complete el efecto.

```
$("#btnOculto").click(function () {
    // Ocultacion lenta.
    $("img").hide("slow",
        function () {
            alert("TERMINADO");
        }
    );
});
```

Las *funciones de retrollamada* pueden indicarse como segundo argumento opcional en las funciones *$().show()*, *$().hide()*, *$().fadeIn()*, *$().fadeOut()*, *$().slideUp()*, *$().slideDown()*.

La función *$().animate()* también permite indicar una función de retrollamada como tercer argumento opcional para que se ejecutare al final.

Ejemplo: El siguiente código muestra una animación sobre múltiples CSS a ejecutarse en 300 milisegundos, tras cuya finalización se muestra el mensaje "TERMINADO".

```
// Animación sobre capa con estilo ".capa"
$("div.capa").animate(
    {
        left: "+=50",       // posicion horizontal +50px respecto posición actual.
        opacity: 0.25       // opacidad al 0.25%
    },
    300,                    // duración de la animación en milisegundos.
     function () {          // función de retro-llamada para cuando termine.
        alert("Terminado")
     }
);
```

Colas de funciones.

Las funciones, efectos y animaciones asignadas en JQuery a uno o varios elementos son gestionados mediante ***colas***. Estas colas que determinan en que orden se ejecutan cada una de las funciones asignadas para cada elemento.

Anteriormente, hemos visto que si deseamos que una acción se ejecute tras un efecto o animación podemos definir una *función de retrollamada*:

```
$("#btnOculto").click(function () {
    // Ocultacion lenta.
    $("img").hide("slow");
    alert("OCULTO");
});
```

Lo mismo puede conseguirse manipulando la cola de funciones del elemento agregando una más mediante la función ***$().queue()***:

```
$("#btnOculto").click(function () {
    // Ocultacion lenta.
    $("img")
        .hide("slow")
        .queue(                    // Se encola la función
            function (next) {
               alert("OCULTO");
               next();   // Ejecuta siguiente función en la cola si la hubiera
            }
        );
});
```

Cuando el elemento ejecuta el filtro, pasa a ejecutar la siguiente función que tiene en cola, la cual en este caso; muestra el mensaje “OCULTO”. El argumento ***next*** representa la siguiente función en la cola a ejecutarse a continuación.

Declaracion de una cola

Es posible declarar y asignar a uno o varios elementos una cola de funciones para que se ejecuten determinadas operaciones en secuencia sobre ellos. Para ello se declaran las diferentes funciones a ejecutar llamando a la función ***$().queue()*** indicando como argumentos el nombre de la cola, y la función que se le agrega.

```
$("img")
    .queue("foo", function (next) {
        alert("funcion 1 de la cola foo");
        next();
    })
    .queue("foo", function (next) {
        alert("funcion 2 de la cola foo");
        next();
    });
```

Cuando se quiera que se ejecuten las funciones de la cola debe llamarse al método ***$().dequeue()*** indicando como argumento el nombre de la cola:

```
$("img").dequeue("foo");
```

Colas de efectos.

Las funciones de fectos devuelven la selección original de elementos sobre la que actúan, lo que permite encadenar diferentes efectos para encadenar diferentes efectos.

Ejemplo: El siguiente código muestra el encadenamiento de un efecto de desaparición de 1 segundo acompañado de un efecto de aparición de medio segundo que se ejecutan en secuencia:

```
$("h1").hide(1000).show(500);
```

También puede emplearse la funcion ***$().delay()*** para esperar un determinado tiempo indicado en milisegundos antes de ejecutar el siguiente efecto en la cadena.

Ejemplo: El siguiente código muestra el encadenamiento de un efecto de desaparición de 1 segundo de duración, seguido de una espera de medio segundo, y terminado con un efecto de reaparición de otro segundo de duración:

```
$("h1").hide(1000).delay(500).show(1000);
```

Animaciones (*$().animate()*)

JQuery permite definir animaciones sobre los atributos CSS partiendo su *valor actual* hasta un *valor final* a lo largo de un determinado *lapso se tiempo*. Para ello se emplea la función ***$().animate()***:

```
$(selector).animate( estilos,
                     Lapso )
```

El argumento *estilos* puede ser uno o varios estilos para los que se indica un valor final. Este valor puede ser absoluto (`opacity: 0.5`), o relativo (`top: "+=25"`). El *lapso* de tiempo indica el tiempo que se desea dure la animación en milisegundos.

Ejemplo: El siguiente código muestra una animación que se desplaza 50 píxels a la derecha a una capa y cambia su opacidad al 25% en 300 milisegundos.

```
// Animación sobre capa con estilo ".capa"
$("div.capa").animate(
    {
        left: "+=50",       // posicion horizontal +50px desde posición actual.
        opacity: 0.25       // opacidad al 0.25%
    },
    300,                    // duración de la animación en milisegundos.
);
```

Ejemplo: El siguiente código muestra una página en la que se muestra una imagen que se desplaza hacia la posicion del puntero del ratón cada vez que el usuario hace clic.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title></title>
    <script src="Scripts/jquery-3.0.0.js"></script>
    <style>
        .imagen {
            position:absolute;
        }
    </style>
    <script>
        $().ready(function () {
            // Cuando el usuario hace clic sobre la página
            $(window).mousedown(function (e) {
                var x = e.pageX;                // Obtencion de posición puntero ratón
                var y = e.pageY;
                var w = $(".imagen").width();   // Obtencion tamaño de la imagen
                var h = $(".imagen").height();
                // Movimiento la imagen a la posicion indicada por el ratón en 100msg
                $(".imagen").animate(
                    {
                        "top": (e.pageY - (h / 2)),  // Cambio posición horizontal
                        "left": (e.pageX - (w / 2))  // Cambio posición vertical
                    },
                    100);
    </script>
</head>
<body>
    <img class="imagen" src="images/imagen0.png" /><br />
</body>
</html>
```

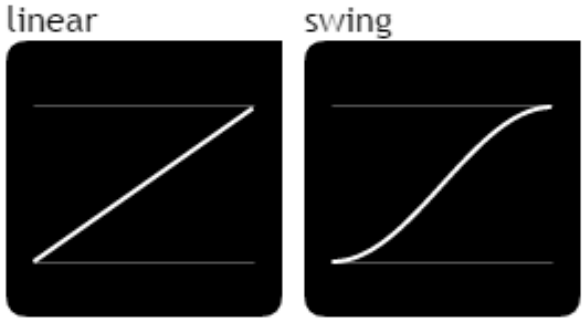
El parámetro *Easing*

Este parámetro indica el modo en que se produce la transición de los valores de los estilos mediante la funcion *$().animate()*. Según como se ajuste permite animaciones más suaves, o que empiecen despacio y se aceleren al final, o viceversa. El parámetro se indica por cada CSS a continuación del valor final:

```
$("div.funtimes").animate(
        {
            left:    ["+=50", "swing"],    // transicion lineal para la propiedad "left"
            opacity: [ 0.25, "linear"]     // transicion swing para la opacidad.
        }
    , 300);
```

JQuery define por defecto los siguientes valores:

- "*lineal*" → Determina una transicion lineal de los valores.
- "*swing*" → Determina una transición suavizada de los valores.



Representacion gráfica de la transición de valores "lineal" y "swing"

Ejercicios

Partiendo del fichero *index.html* de los ejercicios anteriores realizar las siguientes operacione:

1. En la página se muestra una capa con el identificador "blog". Esta capa contiene una serie de párrafos encabezados por un elemento <h3> seguido de un pequeño párrafo de contenido. Se pide que al hacer click en cada uno de los encabezados <h3> se muestre el párrafo correspondiente ocultándose el resto. Por defecto, todos deben permanecer ocultos.

(*) Para ayudarte a seleccionar los elementos visibles e invisibles no olvides que puede emplear los selectores ":visible" y ":hidden".

2. En la página se muestra una barra de navegación con tres opciones. La opción "*Resources*" cuenta con un submenú que no es visible por defecto ya que tiene asignado la regla de estilo *"#nav li ul*" definida en *styles.css* con la propiedad "*display:none*".

Se pide que al posicionarse el puntero del ratón sobre la opción "*Resources*" se muestre mediante JQuery la lista de subopciones y que se oculte al desplazarse el puntero fuera.

3. En la página se muestra un elemento <ul> con el identificador "slideshow". Se pide que mediante JQuery, se muestren cada uno de los elementos <li> contenidos durante un segundo cíclicamente. Así pues, primero debe mostrarse durante 1 segundo el contenido "Fruit", después "Vegetables", después "Bread", y volver a mostrar "Fruit" sucesivamente.

(*) Para ayudarte puede emplear encadenamiento de efectos, haciendo que cada elemento se muestre, espere un segundo, se oculte, e invoque a una función que muestre el siguiente.