

TEMA 04

Integración de Contenido Multimedia.


4.1- Elementos interactivos Básicos y Avanzados.

La interacción es, junto a la inclusión de elementos multimedia, otra de las claves del desarrollo web.

([Guitarra interactiva Google Logos Les Paul](#))

Nos centraremos en la interacción de objetos con JQuery, usando HTML5 y CSS3.

Jquery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol [DOM](#), manejar eventos, desarrollar animaciones....



Jquery es un software libre y de código abierto y su misión es ahorrar tiempo y espacio en desarrollo de animaciones.

Jquery es un lenguaje muy extenso, pero nos centraremos en las interfaces de usuario y la interacción.

4.1.1- Introducción al JQuery.

Para utilizarle tendremos que descargar la librería y enlazarla en la página HTML de la siguiente manera:

```
<script type="text/javascript" src="jquery.js"></script>
```

Se puede descargar en dos versiones:

- **Comprimida(Production)**, recomendable para que la descargue el cliente ya que ocupa menos espacio.
- **Descomprimida(Development)**, adecuada cuando se está trabajando en desarrollo.

jQuery CDN

- Si no desea descargar y hospedar jQuery usted mismo, puede incluirlo desde un CDN (Content Delivery Network).
- Tanto Google como Microsoft alojan jQuery.

Google CDN:

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
</head>
```

Microsoft CDN:

```
<head>  
<script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.3.1.min.js"></script>  
</head>
```

Estructura

```
<html>
  <head>
    <title>      </title>
    <script type="text/javascript"src="jquery-3.3.1.js"></script>
    <script type="text/javascript">

</script>

    <style media="screen">
      /* definimos los estilos de p, body, h1,...
    </style>
  <head>
  <body>

</body>
</html>
```

Una de las primeras instrucciones que todo código JQuery debe tener es:

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

Indica que el código se ejecute cuando el código HTML se ha cargado completamente y el árbol DOM se ha generado. **No espera a que se carguen las imágenes, con los elementos definidos en el HTML es suficiente.**


La función más usada es `$()`, se utiliza para seleccionar un elemento del árbol DOM a través del identificador, del nombre de la clase o de la etiqueta CSS.

```
$("#miidentificador");
```

```
$("a")      ➡      enlace
```

```
$("#miparrafo, a");      ➡      varios al mismo tiempo.
```

```
$(".miclase");
```



Con `$(document)` se obtiene una referencia al documento (la página web) que se está cargando.

Luego, con el método `ready()` se define un evento, que se desata al quedar listo el documento para realizar acciones sobre el DOM de la página.

4.1.2- Manejo de eventos

Además de `ready()`, JQuery dispone de varias funciones relacionadas con la gestión de eventos.

Para cada evento hay dos posibilidades de manejo:

1- Se le puede pasar la función que determine su comportamiento.

```
$(“p”).click(function() { alert($(this).text());});
```

Este código muestra una ventana de alerta cuando se hace click en cualquier párrafo del documento.

2- No se le pasa la función, entonces se ejecuta el evento JavaScript del elemento. El nombre de cada función es el mismo que el del evento, pero sin el prefijo “on” de los eventos.

```
$(“p”).click();
```

Se ejecuta la función `click()` que es la asociada por defecto al evento `onclick`.

Bind() y unbind()

Un método interesante para trabajar con eventos de manera óptima es bind(). Permite asociar a un elemento un número ilimitado de eventos, todos de una vez.

```
bind("tipo_de_evento1 tipo_de_evento2...tipo_de_evento  
n",manejador(handler))
```

Ejemplo

```
$("#p").bind("click mouseenter mouseleave", function(e))
```

Existe el método unbind(), que elimina un evento asignado a un elemento o varios de una página.

Ejemplo

```
$("#p").unbind("click");
```

Ejercicio 1.

Escribe el código correspondiente que averigua si todos los elementos de un párrafo son de un color (rojo) al hacer click sobre el, o entra o sale con el ratón. Si no es de color **rojo** entonces se lo asigna ,y si lo es entonces lo cambia a **azul**.

Para no eliminar todo el manejo de eventos de un tipo determinado se puede especificar la función que se desea descartar en la lista de parámetros de la llamada a **unbind()**.

Para ello habría que colocar la función dentro de una variable, para poder referirse a ella como parámetro de **unbind()**.

Ejercicio 2.

Quita la atención al evento onclick de p del ejercicio anterior, pero permite que se pueda seguir cambiando el color de p.

4.2- Cambio en las propiedades de un elemento.

4.2.1- Cambio de las propiedades CSS

Si se desea obtener el valor de una propiedad para un elemento determinado se utiliza el método `.css()`

```
$("#p").css("color");
```

Expresa con que color está definido los elementos de <p>

Permite establecer el valor de una propiedad determinada.

```
$("#p").css("color", "red");
```

Cambia el color de la propiedad color de los elementos de <p>.

También permite establecer varias propiedades a la vez.

```
$("#p").css({color: "red", background: "blue", font-weight: "bold"});
```

4.2.2- Añadir nuevos elementos

`append()` permite insertar un contenido como el último hijo (árbol DOM) de cada elemento de la colección seleccionada.

Si lo que se desea es insertarlo como primer hijo, entonces se usa `.prepend()`.

Ejercicio 4.*

Añade una etiqueta `<p>` dentro de dos `<div>` llamados “misdivs”.

Ejercicio 5.

Crea el código correspondiente para que se añada una etiqueta <H1>, que ya existe, al interior de dos <div> al hacer click con el ratón.

4.2.4- Otros métodos.

`.innerWidth()` e `.innerHeight()`: Para un elemento que devuelve sus dimensiones internas (anchura y altura) contando el padding del elemento pero no el borde.

`.outerWidth()` y `.outerHeight()`: Devuelve las dimensiones externas (anchura y altura) del elemento contando el padding y su borde.

`.offset`: Indica la posición del elemento real, teniendo en cuenta los márgenes del elemento. Lo devuelve en un objeto con dos atributos `top` y `left`.

4.3- Ejecución de secuencia de funciones.

Uno de los aspectos más utilizados en Jquery son los llamados **callbacks**. Cuando se trabaja de cara a la interfaz, es habitual utilizar secuencia de métodos o funciones apilados para lograr efectos más elaborados.

Si se desea que unos<div> con identificador “#misdivs” se desvanezca tardando 2 segundos, luego cambie de color y termine apareciendo de nuevo en 2 segundos, se podría usar:

```
$("#misdivs").fadeOut (2000);  
$("#misdivs").css ({background: "blue"});  
$("#misdivs").fadeIn (2000);
```

Lo podemos expresar en una sola secuencia:

```
$("#misdivs").fadeOut(2000).css({background:"blue"}).fadeIn(2000);
```


Para las dos sintaxis el problema es el mismo: para JQuery todo se hace al mismo tiempo.

La solución es utilizar efectos `.delay()` o utilizar la pila de ejecución de funciones.

Cualquier método o función en JQuery puede recibir como parámetro la función que se ejecutará después de ejecutar el método o función correspondiente. Esta función se llama callback.

En el ejemplo anterior se puede ejecutar el primer desvanecimiento, y en el callback de ese método colocar el cambio de propiedad con `.css()` y que aparezca el nuevo elemento.

Definimos la función:

```
var ocultar=function () {  
    $("misdivs").css({background: "green"});  
    $("misdivs").fadeIn(2000);  
};
```

La llamada sería:

```
$("misdivs").fadeOut(2000, ocultar);
```

4.4- Comportamiento de los elementos. Efectos visuales

4.4.1- Efectos visuales

Con jQuery, se puede ocultar y mostrar elementos HTML con los métodos `hide ()` y `show ()`:

Sintaxis:

```
$( selector ).hide( speed,callback );
```

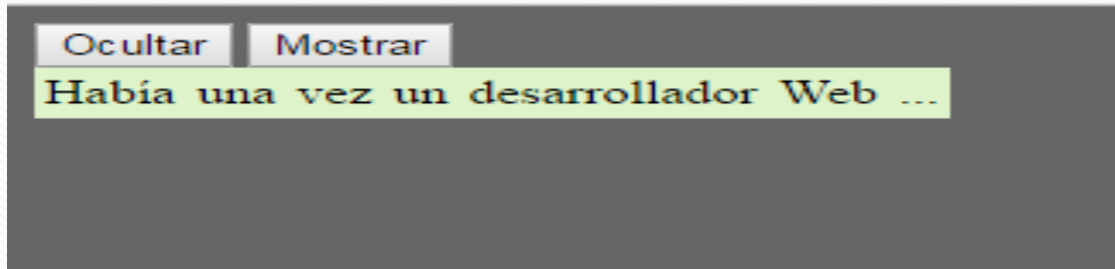
```
$( selector ).show( speed,callback );
```

El parámetro de velocidad (`speed`) es opcional y especifica la velocidad de la ocultación / visualización. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

Ejercicio 6

Realiza el código necesario para que poniendo dos botones realicen:

- 1.- Mostrar el texto(span).
- 2.- Ocultar el texto(span).
- 3.- La velocidad para ocultarlo será de 500 ms.



4.4.2- Efectos de opacidad-transparencia de los elementos

El método jQuery `fadeIn ()` se utiliza para mostrar un elemento oculto.

Sintaxis:

```
$( selector ).fadeIn( speed,callback );
```

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el atenuado.

Ejemplo

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```

El método jQuery `fadeOut ()` se utiliza para ocultar un elemento visible.

Sintaxis:

`$(selector).fadeOut(speed,callback);`

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el atenuado.

Ejemplo

```
$("#button").click(function(){  
    $("#div1").fadeOut();  
    $("#div2").fadeOut("slow");  
    $("#div3").fadeOut(3000);  
});
```

El método jQuery `fadeToggle ()` cambia entre los métodos `fadeIn ()` y `fadeOut ()`.

Si los elementos se ocultan, `fadeToggle ()` los mostrará.

Si los elementos se muestran, `fadeToggle ()` los eliminará.

Sintaxis:

`$(selector).fadeToggle(speed,callback);`

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el atenuado.

Ejemplo

```
$("#button").click(function(){  
    $("#div1").fadeToggle();  
    $("#div2").fadeToggle("slow");  
    $("#div3").fadeToggle(3000);  
});
```

El método jQuery `fadeTo ()` permite atenuar (disminuir) una opacidad dada (valor entre 0 y 1).

Sintaxis:

`$(selector).fadeTo(speed,opacity,callback);`

El parámetro de **velocidad** es obligatorio y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de **opacidad** es obligatorio en el método `fadeTo ()` especifica el atenuado a una opacidad dada (valor entre 0 y 1).

El parámetro de devolución de llamada opcional es una función que se ejecutará una vez finalizada la función.

Ejemplo

```
$("#button").click(function(){  
    $("#div1").fadeOut("slow", 0.15);  
    $("#div2").fadeOut("slow", 0.4);  
    $("#div3").fadeOut("slow", 0.7);  
});
```

Obtener contenido.

Tres métodos simples, pero útiles, jQuery para la manipulación de DOM son:

- **Text ()** - Establece o devuelve el contenido de texto de los elementos seleccionados
- **Html ()** - Establece o devuelve el contenido de los elementos seleccionados (incluido el marcado HTML)
- **Val ()** - Establece o devuelve el valor de los campos de formulario

Manipulando CSS

jQuery tiene varios métodos para manipular CSS. Examinaremos los siguientes métodos:

- **AddClass ()** - Agrega una o más clases a los elementos seleccionados
- **RemoveClass ()**: elimina una o más clases de los elementos seleccionados
- **ToggleClass ()** - Alterna entre agregar / quitar clases de los elementos seleccionados
- **Css ()** - Establece o devuelve el atributo de estilo

Ejercicio 7

- En nuestro ejemplo vamos a tener dos elementos. Primero una capa DIV con un texto. Después tendremos un enlace que estará fuera de la capa DIV. Al situar el usuario el ratón sobre un enlace añadiremos una clase CSS a la capa DIV y al retirar el ratón del enlace eliminaremos la class CSS que habíamos añadido antes.

```
<html>
<head>
  <title>Añadir y quitar clases CSS a elementos</title>
  <script src=" " type="text/javascript"></script>
  <style type="text/css">
    .clasecss{
      background-color: #ff8800;
      font-weight: bold;
    }
  </style>
</head>

<body>
<div id="capa">
Esta capa es independiente y voy a añadir y eliminar clases css sobre ella
</div>
<br>
<br>
<a href="https://desarrolloweb.com">Añadir y quitar clase en la capa de arriba</a>
</body>
</html>
```

Ejemplo de hoja de estilo.

La siguiente hoja de estilo se utilizará para todos los ejemplos de esta página:

```
.important {  
    font-weight: bold;  
    font-size: xx-large;  
}
```

```
.blue {  
    color: blue;  
}
```

Ejemplo addClass.

```
$("#button").click(function(){  
    $("#h1, h2, p").addClass("blue");  
    $("#div").addClass("important");  
});
```

También puede especificar **varias clases dentro del método** addClass ():

```
$("#button").click(function(){  
    $("#div1").addClass("important blue");  
});
```


Ejemplo removeClass

```
$("#button").click(function(){  
    $("#h1, h2, p").removeClass("blue");  
});
```

Ejemplo toggleClass

```
$("#button").click(function(){  
    $("#h1, h2, p").toggleClass("blue");  
});
```

Método `css()`

El método `css ()` establece o devuelve una o más propiedades de estilo para los elementos seleccionados.

Para **devolver** el valor de una propiedad CSS especificada, utilice la siguiente sintaxis:

```
css(" propertyname ");
```

Ejemplo

```
$("p").css("background-color");
```

Para **establecer** una propiedad CSS especificada, utiliza la siguiente sintaxis:

Permite también establecer varias a la vez.

*css(" **propertyname** ", " **value** ");*

Ejemplo

```
$("p").css("background-color", "yellow");
```

```
$("p").css({"background-  
color": "yellow",  
"font-size": "200%"});
```

Ejercicio 8*

Realiza el código necesario para que en un texto se seleccionen los adjetivos y al hacer click sobre ellas desaparezcan y se muestra un mensaje en un <div>(inicialmente vacío).

Encuentra los adjetivos

La chaqueta roja el lo mejor cuando hace un tiempo cálido ¿verdad?

Encuentra los adjetivos 'roja' ¡Se ha desvanecido!

La chaqueta el lo mejor cuando hace un tiempo cálido ¿verdad?

4.4.3- Efectos de Desplazamiento

El método jQuery `slideDown()` se utiliza para deslizar hacia abajo un elemento.

Sintaxis:

`$(selector).slideDown(speed, callback);`

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el deslizamiento.

Ejemplo

```
$("#flip").click(function(){  
  $("#panel").slideDown();  
});
```



El método jQuery `slideUp ()` se utiliza para deslizar hacia arriba un elemento.

Sintaxis:

`$(selector).slideUp(speed, callback);`

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el deslizamiento.

Ejemplo

```
$("#flip").click(function(){  
  $("#panel").slideUp();  
});
```

El método jQuery `slideToggle ()` alterna entre los métodos `slideDown ()` y `slideUp ()`.

Si los elementos se han deslizado **hacia abajo**, `slideToggle ()` se deslizará **hacia arriba**.

Si los elementos se han deslizado **hacia arriba**, `slideToggle ()` los deslizará **hacia abajo**.

`$(selector).slideToggle(speed,callback);`

El parámetro de velocidad es opcional y puede tomar los siguientes valores: "lento", "rápido", milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el deslizamiento.

Ejemplo

```
$("#flip").click(function(){  
  $("#panel").slideToggle();  
});
```

Ejercicio 9*

Realiza el código correspondiente para que se muestren 4 <div>. Tres de ellos tienen texto y el otro tiene un botón.

- 1.- Al hacer click en el texto ¡Pulsa Aquí! Se despliegan los <div>.
 - 2.- Al hacer otra vez se vuelven a contraer (hide()).
- ¡¡¡Os doy el <style>!!!

<style>

```
div { background:#de9a44; margin:3px; width:80px;
height:140px; display:none; float:left; }
```

</style>

Los efectos
no son sólo
para divs o
spans

También
son para
cualquier
elemento

Botones
también

aunque no
estén
incluidos en
un div

¡Pincha Aquí!

4.4.4- Otros Efectos.

El método jQuery `animate ()` se utiliza para crear animaciones personalizadas.

Sintaxis:

```
$( selector ).animate({ params },speed,callback );
```

El parámetro `params` es requerido y define las propiedades CSS que se van a animar.

El parámetro `speed` es opcional y especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará una vez finalizada la animación

El ejemplo siguiente demuestra un uso simple del método `animate()`;

Mueve un elemento `<div>` a la derecha, hasta que haya alcanzado una propiedad `left` de `250px`:

Ejemplo

```
$("#button").click(function(){  
    $("#div").animate({left: '250px'});  
});
```

De forma predeterminada, todos los elementos HTML tienen una posición estática y no se pueden mover. Para manipular la posición, recuerde primero establecer la propiedad de posición CSS del elemento a relativa, fija o absoluta.

Se pueden animar varias propiedades al mismo tiempo.

Ejemplo

```
$("#button").click(function(){  
  $("#div").animate({  
    left: '250px',  
    opacity: '0.5',  
    height: '150px',  
    width: '150px'  
  });  
});
```

¡¡¡¡¡IMPORTANTE!!!!

¿Es posible manipular TODAS las propiedades CSS con el método animate ()?

Casi. Sin embargo, hay una cosa importante que recordar: todos los nombres de propiedad deben ser **camel-cased** cuando se utiliza con el método animate (): Tendrá que escribir paddingLeft en lugar de padding-left, marginRight en lugar de margin-right, y así sucesivamente.

Además, la animación de color no se incluye en la biblioteca central de jQuery.

Si quieres animar el color, necesitas descargar el complemento Color Animations de jQuery.com.

También es posible definir valores relativos (el valor es entonces relativo al valor actual del elemento). Esto se hace poniendo + = o - = delante del valor:

Ejemplo

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});
```

El método **delay ()** establece un temporizador para retrasar la ejecución del siguiente elemento de la cola.

Sintaxis

\$(selector).delay(speed,queueName)

Speed es opcional y especifica la velocidad del retraso. Los valores son: milisegundos, fast, slow.

queueName es opcional y especifica el nombre de la cola que contiene los elementos que se retrasarán. Su valor por defecto es “fx” y es el efecto estándar de la cola.

Ejercicio 10

Escribe el código correspondiente para que al hacer click sobre cualquiera de los párrafos, se recogen todos (slideUp(300)). Aparecen (fadeIn(800)) los que tienen identificador “antes” y más tarde (delay(800)) los que tienen identificador “después”.

```
<style>
```

```
div.antes { background-color:#aac; }
```

```
div.despues{ background-color:#abc;}
```

```
</style>
```

Pincha en cualquiera de los textos

Antes

Después

Antes

Después