

***RECUERDA PONER A GRABAR LA
CLASE***





¿DUDAS DEL ON-BOARDING?

MIRALO AQUI



Clase 13. JAVASCRIPT

EFFECTOS Y ANIMACIONES CON JQUERY



OBJETIVOS DE LA CLASE

- Aprender a usar efectos de animación de jQuery, y combinarlos entre sí.
- Entender cómo modificar el CSS desde jQuery.

GLOSARIO:

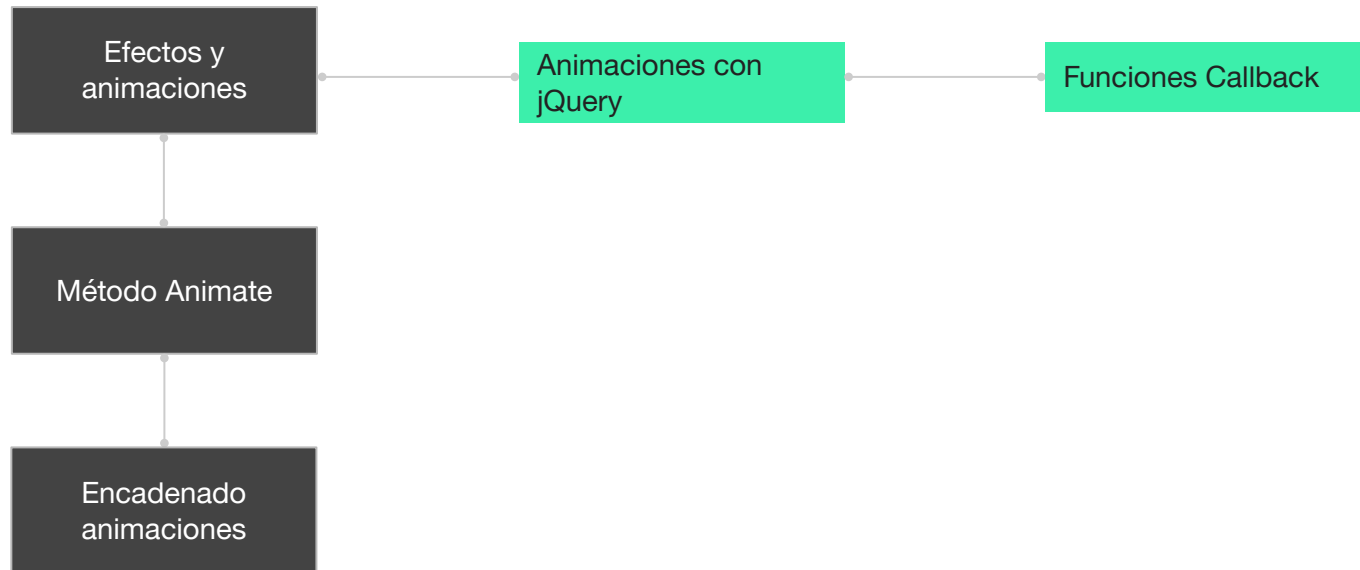
Clase 12

Selectores en jQuery: se refiere a la forma de identificar a los diferentes elementos del DOM y poder operar sobre ellos. Usando un selector, se puede ocultar un elemento, agregar uno nuevo al DOM, animarlo, o también realizar llamadas mediante AJAX. Es decir que el selector, es la herramienta más importante de jQuery.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 13

¡Para
recordar!



CRONOGRAMA DEL CURSO

Clase 12



jQuery: selectores y eventos



EJEMPLOS EN VIVO



INCORPORAR JQUERY AL PROYECTO

Clase 13



Efectos y animaciones con jQuery



EJEMPLOS EN VIVO



ANIMACIONES EN JQUERY

Clase 14



AJAX con jQuery



AJAX EN TU PROYECTO



TERCERA ENTREGA DEL PROYECTO FINAL



HERRAMIENTAS DE LA CLASE

Les compartimos algunos recursos para acompañar la clase

- Guión de clase N° 13 [aquí](#).
- Quizz de clase N° 13 [aquí](#)
- Booklet de Javascript [aquí](#)
- FAQs de Javascript [aquí](#)

EFFECTOS Y ANIMACIONES

ANIMACIONES CON JQUERY



JQuery provee métodos para incorporar efectos y animaciones a los elementos del DOM.

El objetivo es combinar las diferentes variantes entre sí para darle a una web o app dinamismo y movimiento.

SHOW

El método `show()` muestra un elemento del DOM. Suponiendo que se encuentre oculto mediante css (`display: none`), al seleccionarlo con JQuery y ejecutar `show()`, el elemento aparecerá en la pantalla.

```
//Agregamos <h3> con jquery ocultos con style="display: none;"
$("body").prepend('<h3 style="display: none" >¡Hola Coder1!</h3>');
$("body").prepend('<h3 style="display: none" >¡Hola Coder2!</h3>');
//Mostramos con show() todos los <h3>
$("h3").show();
```

HIDE

El método `hide()` es la inversa a `show`. Suponiendo que se encuentre visible mediante `css`, al seleccionarlo con JQuery y ejecutar `hide()`, el elemento desaparecerá de la pantalla.

```
//Agregamos <h3> con jquery
$("body").prepend('<h3>Hola Coder1!</h3>');
$("body").prepend('<h3>Hola Coder2!</h3>');
//Ocultamos con hide() todos los <h3>
$("h3").hide();
```

FADE IN

El método `fadeIn()` tiene el mismo fin que la función `show()`, solo que en este caso, el elemento aparece con una transición. Es decir, que al llamar `fadeIn()` el elemento empieza a aparecer de forma gradual, hasta que esté completamente visible.

```
//Agreguemos <h3> con jQuery ocultos con style="display: none;"
$("body").prepend('<h3 style="display: none" >¡Hola Coder1!</h3>');
$("body").prepend('<h3 style="display: none" >¡Hola Coder2!</h3>');
//Mostramos con fadeIn() todos los <h3>
$("h3").fadeIn();
```

FADE OUT

El método `fadeOut()` tiene el mismo fin que la función `hide`, solo que en este caso, el elemento desaparece con una transición. Es decir, que al llamar `fadeOut` el elemento empieza a desaparecer de forma gradual, hasta que esté completamente invisible.

```
//Agregemos <h3> con jQuery"
$("body").prepend('<h3>¡Hola Coder1!</h3>');
$("body").prepend('<h3>¡Hola Coder2!</h3>');
//Ocultamos con fadeOut() todos los <h3>
$("h3").fadeOut();
```

VELOCIDAD DEL FADE

Tanto para `fadeOut()` como `fadeIn()` se puede enviar un parámetro opcional para indicar la velocidad el fade. Se puede usar la palabra "slow" o "fast", o también un número en milisegundos para indicar la duración total de la transición.

Fade In



`.fadeIn("slow")`

CALLBACK DEL FADE

Tanto para `fadeOut()` como `fadeIn()` se le puede enviar un segundo parámetro opcional para indicar una función que se ejecutará cuando la animación termine.

```
//Agregamos <h3> con jQuery
$("body").prepend('<h3>¡Hola Coder1!</h3>');
//Ocultamos con fadeOut() todos los <h3>
$("h3").fadeOut("slow", function(){
    //Cuando termina de ocultarse el elemento lo mostramos nuevamente
    $("h3").fadeIn(1000);
});
```

SLIDE DOWN

El método `slideDown()` hace aparecer al elemento haciendo una transición hacia abajo. Es una alternativa al `fadeIn()`. Se puede enviar los mismos parámetros de velocidad y callback que en `fadeIn` y `fadeOut`.

```
//Agregamos un botón y un div con jQuery
$("body").prepend('<button id="btn1">Mostrar</button>');
$("body").prepend('<div id="div1" style="height: 120px">
    <h3>¡Hola Coder!</h3>
    <h4>Sorpresa 2</h4>
</div>');

//Usamos slideDown sobre div1 en respuesta al click del boton btn1
$("#btn1").click(() => {
    $("#div1").slideDown("fast");
});
```

SLIDE UP

El método `slideUp()` hace desaparecer al elemento haciendo una transición hacia arriba. Es una alternativa al `fadeOut`. Se puede enviar los mismos parámetros de velocidad y callback que en `fadeIn` y `fadeOut`.

```
//Agregamos un botón y un div con jQuery
$("body").prepend('<button id="btn1">Mostrar</button>');
$("body").prepend('<div id="div1" style="height: 120px">
    <h3>¡Hola Coder!</h3>
    <h4>Sorpresa 2</h4>
</div>`);

//Usamos slideUp sobre div1 en respuesta al click del boton btn1
$("#btn1").click(() => {
    $("#div1").slideUp("fast");
});
```

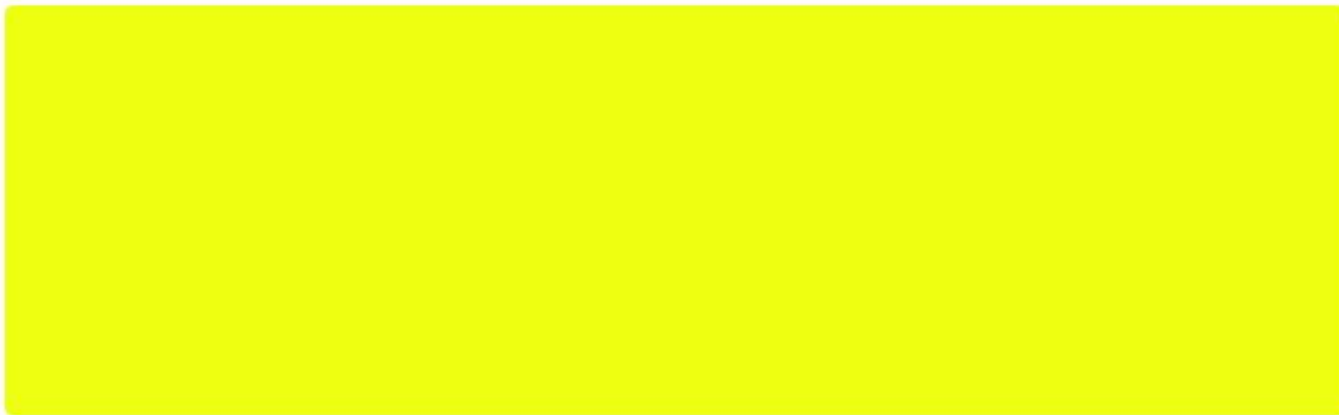
SLIDE TOGGLE

El método `slideToggle()` hace aparecer o desaparecer el elemento mediante `slide`. En función de que el elemento esté visible o no, ejecuta `slideUp`, o `slideDown`.

```
//Agregamos un botón y un div con jQuery
$("body").prepend('<button id="btn1">Mostrar</button>');
$("body").prepend(`<div id="div1" style="height: 120px">
    <h3>¡Hola Coder!</h3>
    <h4>Sorpresa</h4>
</div>`);

//Usamos toggle sobre div1 en respuesta al click del botón btn1
$("#btn1").click(() => {
    $("#div1").toggle("fast");
});
```

EJEMPLO



slideUp

slideDown

slideToggle

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!

CODER HOUSE



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

**¿Estás terminando el curso y te gustaría
aumentar tu empleabilidad?**

**Recordá que desde del área de talento
podemos ayudarte.**

Conoce todos nuestros servicios [aquí](#)

¡5/10 MINUTOS Y VOLVEMOS!

CODER HOUSE

MODIFICAR CSS EN JQUERY

MODIFICAR CSS

jQuery provee un método para aplicar cambios directos sobre el css del elemento. Llamando a `css()` podemos indicar la propiedad a modificar del elemento seleccionado.

La sintaxis funciona así: `selector.css("propiedad-css","valor");`

```
//Agregamos un parrafo con jQuery
$("body").prepend('<p class="titulo">Code House</p> ');
//Modificamos las reglas CSS desde jQuery
$("p").css("background-color", "yellow");
$("p").css("width", "50%");
$(".titulo").css({  "color": "#ccc",
                    "font-size": "40px",
                    "borderLeft": "5px solid #ccc" });
```

MÉTODO ANIMATE

ANIMATE

El método `animate()` nos permite crear efectos de animación sobre cualquier propiedad CSS numérica. Se requiere como parámetro un objeto de propiedades CSS. Este objeto es similar al que se puede enviar al método `.css()`, salvo que el rango de propiedades es más restrictivo.

```
//Agregamos un parrafo con jQuery
$("body").prepend('<p class="titulo">Code House</p>');
//Animamos sus propiedades CSS con animate
$(".titulo").animate({ left:'250px',
                        opacity:'0.5',
                        height:'150px',
                        width:'150px' }, //1er parámetro propiedades
                      "slow",           //2do parámetro duración
                      function() {      //3er parámetro callback
                        console.log("final de animación");
                      });
```

EJEMPLO APLICADO: SCROLL ANIMADO

```
//Agregamos una estructura con jQuery
$("body").prepend(`<div>
    <a>Ir a contacto</a>
    <p style="height: 800px"></p>
    <section id="seccionContacto">
        <h4>¡Somos Coders!</h4>
    </section>
</div>`);

// Asociamos la animación al click en un elemento <a>
$('a').click( function(e) {
    e.preventDefault();
    //Animamos sus propiedades CSS con animate
    $('html, body').animate({
        scrollTop: $("#seccionContacto").offset().top
    }, 2000);
} );
```

ENCADENAR ANIMACIONES

ENCADENAR ANIMACIONES

Es posible encadenar métodos jQuery, lo cual permite definir una animación tras otra empleando en una única declaración.

Es recomendable emplear esta forma si es necesario asociar múltiples animaciones consecutivas a un mismo elemento. Para ahorrarnos consultas al DOM.

```
//Agreguemos un párrafo con jQuery
$("body").prepend('<p id="p1">Coder House</p>');
//Declaración de métodos encadenados
$("#p1").css("color", "red")
        .slideUp(2000)
        .slideDown(2000);
```

MÉTODO DELAY

El método `delay()` nos permite retrasar la ejecución de comandos encadenados. Puede utilizarse para establecer un tiempo de espera en milisegundos entre una animación y la siguiente.

```
//Agregamos un párrafo con jQuery
$("body").prepend('<p id="p1">Coder House</p>');
//Declaración de métodos encadenados
$("#p1").css("color", "red")
        .slideUp(2000)
        .delay(2000)
        .slideDown(2000);
```


EJEMPLO APLICADO: ANIMANDO LA COMPRA

```
// Array de objetos para agregar información al DOM.
const productos = [{ id: 1, nombre: "Arroz", precio: 125 },
{ id: 2, nombre: "Fideo", precio: 70 },
{ id: 3, nombre: "Pan" , precio: 50},
{ id: 4, nombre: "Flan" , precio: 100}];

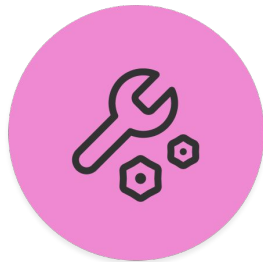
// Asociamos el evento click en ready luego del DOM Generado
$(document).ready(function () {
    $(".btnComprar").click(function (e) {
        //Obtenemos hijos del padre <div> desde el target
        let hijos = $(e.target).parent().children();
        //Primer input, valor de ID oculto
        console.log(hijos[0].value);
        //Animación de respuesta de compra
        $(e.target).parent().slideUp("slow");
    });
});

// Recorremos el array con for..of
for (const producto of productos) {
    //Por cada producto ademas de los datos agregamos un botón
    $("#app").append(`<div>
        <input value="${producto.id}" type="hidden">
        <h4> Producto: ${producto.nombre}</h4>
        <b> $ ${producto.precio}</b>
        <button class="btnComprar">Comprar</button>
    </div>`);
}
```

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!



ANIMACIONES EN JQUERY

Desarrolla un algoritmo utilizando HTML, JS y JQuery con animaciones.



VARIANTES DEL EJERCICIO

Elimina un elemento usando una animación de jQuery:

El método `removeElementWithAnimation()` tiene que recibir como parámetro un Id y eliminar el elemento, usando en conjunto una animación de jQuery para que el cambio no sea tan brusco.

Cuentas con 20 minutos para realizar la actividad.



ANIMACIONES ENCADENADAS

Crea animaciones concatenadas sobre uno o más elementos.

ANIMACIONES CONCATENADAS

Formato: código fuente en JavaScript

[Sublime Text](#) o [VisualStudio](#).

Sugerencia: ten en cuenta que los métodos de jQuery show, hide, fadeIn, fadeOut, slideUp, slideDown y toggle ofrecen animaciones con comportamiento predefinido.

Si quieres armar tus propias animaciones, deberás usar animate.

También recuerda que los métodos jQuery pueden encadenarse:

<http://www.w3big.com/es/jquery/jquery-chaining.html>

Desafío
Complementario



>> Consigna: codifica animaciones concatenadas sobre uno o más elementos. Es decir que luego de finalizar una animación en su función callback, se especifica la llamada a otra animación.

>>Aspectos a incluir en el entregable:

Archivo HTML y archivo JavaScript referenciado, que incluya la definición de dos o más animaciones y sus respectivas funciones callback.

>>Ejemplo de secuencia de animación:

- 1) Mostrar un elemento con fadeIn() y al concluir su transición, ocultar otro elemento con fadeOut().
- 2) Aumentar el margen de un elemento con animate() y al concluir su transición, disminuir el margen del mismo elemento con animate().
- 3) Disminuir la altura de un elemento con animate() y al concluir su transición, esperar con delay() unos ms, y volver al tamaño original.

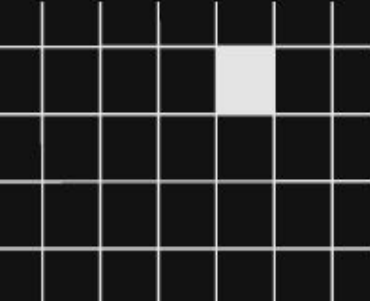
¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Animaciones en JQuery.
 - Método Animate
 - Encadenando Animaciones
- 



OPINA Y VALORA ESTA CLASE