

Ambientes no propietarios

jQuery

Edwin Salvador

15 de diciembre de 2015

Sesión 11

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Recomendaciones

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.

Recomendaciones

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>

Recomendaciones

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>
- En el repo tenemos un libro “Introducción a CSS”

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>
- En el repo tenemos un libro “Introducción a CSS”
- CSS3 cheat sheet

Recomendaciones

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>
- En el repo tenemos un libro “Introducción a CSS”
- CSS3 cheat sheet
- jQuery cheat sheet <http://oscarotero.com/jquery/>

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>
- En el repo tenemos un libro “Introducción a CSS”
- CSS3 cheat sheet
- jQuery cheat sheet <http://oscarotero.com/jquery/>
- Apendice B del libro “Learning PHP, MySQL & JavaScript, 4th Edition”

- Al desarrollar aplicaciones web es 200 % recomendable mantener la herramienta de depuración del navegador siempre abierta para detectar errores rápidamente.
- Practicar los selectores CSS es importante para poder aprovechar otras herramientas como jQuery. <http://strml.net/>
- En el repo tenemos un libro “Introducción a CSS”
- CSS3 cheat sheet
- jQuery cheat sheet <http://oscarotero.com/jquery/>
- Apendice B del libro “Learning PHP, MySQL & JavaScript, 4th Edition”
- Apendice E del mismo libro: selectores, objetos y funciones de jQuery.

Contenido I

- 1 Recomendaciones
- 2 **Introducción**
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:

<https://www.similartech.com/categories/javascript>

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
<https://www.similartech.com/categories/javascript>
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
<https://www.similartech.com/categories/javascript>
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.
- Existen muchos plugins que están basados en jQuery que brindan más utilidades para los desarrolladores web.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
<https://www.similartech.com/categories/javascript>
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.
- Existen muchos plugins que están basados en jQuery que brindan más utilidades para los desarrolladores web.
- Licencia MIT: permite realizar cualquier tipo de trabajos con jQuery prácticamente sin restricciones mientras se mantenga la cabecera de copyright intacta.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
`https://www.similartech.com/categories/javascript`
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.
- Existen muchos plugins que están basados en jQuery que brindan más utilidades para los desarrolladores web.
- Licencia MIT: permite realizar cualquier tipo de trabajos con jQuery prácticamente sin restricciones mientras se mantenga la cabecera de copyright intacta.
- La versión que utilizemos dependerá del tipo de navegadores que queramos soportar. Las versiones 2.X no soportan versiones anteriores a IE9.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
`https://www.similartech.com/categories/javascript`
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.
- Existen muchos plugins que están basados en jQuery que brindan más utilidades para los desarrolladores web.
- Licencia MIT: permite realizar cualquier tipo de trabajos con jQuery prácticamente sin restricciones mientras se mantenga la cabecera de copyright intacta.
- La versión que utilizemos dependerá del tipo de navegadores que queramos soportar. Las versiones 2.X no soportan versiones anteriores a IE9.
- Versión comprimida o editable? Mejor elegir al versión comprimida para sitios en producción por rendimiento.

Introducción

- jQuery es una librería JS que brinda compatibilidad entre navegadores.
- Es la más utilizada de su categoría:
`https://www.similartech.com/categories/javascript`
- Permite manipular el DOM, funciones especiales, manipular el CSS, eventos, efectos animaciones, y una de las mas importantes: **Ajax**.
- Existen muchos plugins que están basados en jQuery que brindan más utilidades para los desarrolladores web.
- Licencia MIT: permite realizar cualquier tipo de trabajos con jQuery prácticamente sin restricciones mientras se mantenga la cabecera de copyright intacta.
- La versión que utilizemos dependerá del tipo de navegadores que queramos soportar. Las versiones 2.X no soportan versiones anteriores a IE9.
- Versión comprimida o editable? Mejor elegir al versión comprimida para sitios en producción por rendimiento.
- `http://jquery.com/`

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Incluyendo jQuery

En un servidor local:

```
<script src="http://myserver.com/jquery-1.11.1.min.js"></script>
```

En un servidor remoto o CDN (Content Delivery Network). Si están seguros que los clientes tendrán acceso a Internet, esta es la forma más sugerida de incluir jQuery.

```
<script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Obtener siempre la ultima version -->
<script src='http://code.jquery.com/jquery-latest.min.js'></script>
<!-- Servidores de Microsoft -->
<script src='http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.1.min.js'></script>
<!-- Servidores de Google -->
<script src='http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js'>
</script>
```

Incluyendo jQuery

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Demo</title>
</head>
<body>
  <a href="http://jquery.com/">jQuery</a>
  <script src="jquery.js"></script>
  <script>

    // Your code goes here.

  </script>
</body>
</html>
```

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready**
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Document Ready

- Para asegurarnos que el código jQuery se ejecute cuando todo el documento HTML se haya cargado debemos escribir todo nuestro código dentro de la función:

```
$( document ).ready(function() {  
    // Your code here.  
});
```

Document Ready

- Para asegurarnos que el código jQuery se ejecute cuando todo el documento HTML se haya cargado debemos escribir todo nuestro código dentro de la función:

```
$( document ).ready(function() {  
    // Your code here.  
});
```

- Existe la versión en JS puro pero está no permite la ejecución hasta que se hayan cargado todas las imágenes (jQuery solo toma en cuenta los elementos HTML)

```
window.onload = function() {  
    alert( "welcome" );  
};
```


Ejemplo

Document Ready

```
$( document ).ready(function() {  
    $( "a" ).click(function( event ) {  
        alert( "Gracias por visitarnos." );  
  
        // Si quisieramos prevenir el comportamiento por  
        // defecto del elemento HTML podemos utilizar:  
        // event.preventDefault();  
    });  
});
```

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis**
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Sintaxis

```
// Elemento
$('p').css('font-family', 'monospace')
$('code').css('border', '1px solid #aaa')

// ID del elemento
color = $('#elem').css('color')

// Clase del elemento
$('.new').css('text-decoration', 'underline')

// Combinados
$('blockquote, #advert, .new').css('font-weight', 'bold')
```

- La función de jQuery `.addClass('clase')` añade una clase CSS al elemento seleccionado.

Ejercicio

- La función de jQuery `.addClass('clase')` añade una clase CSS al elemento seleccionado.
- La función `.removeClass('clase')` la elimina.

Ejercicio

- La función de jQuery `.addClass('clase')` añade una clase CSS al elemento seleccionado.
- La función `.removeClass('clase')` la elimina.
- Escribir código HTML y jQuery y CSS que permita añadir y quitar clases a diferentes elementos HTML.

Ejercicio

- La función de jQuery `.addClass('clase')` añade una clase CSS al elemento seleccionado.
- La función `.removeClass('clase')` la elimina.
- Escribir código HTML y jQuery y CSS que permita añadir y quitar clases a diferentes elementos HTML.
- ¿Qué hace la función `.toggleClass('hide')`?

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM**
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Contenido de un elemento del DOM

- Podemos tomar el contenido de un elemento del DOM (`innerHTML`) utilizando `.html()`

Contenido de un elemento del DOM

- Podemos tomar el contenido de un elemento del DOM (`innerHTML`) utilizando `.html()`
- Para modificarlo `.html('nuevo contenido HTML')`

Contenido de un elemento del DOM

- Podemos tomar el contenido de un elemento del DOM (`innerHTML`) utilizando `.html()`
- Para modificarlo `.html('nuevo contenido HTML')`
- Podemos utilizar también `.text()` o `.text('nuevo texto')`

Contenido de un elemento del DOM

- Podemos tomar el contenido de un elemento del DOM (`innerHTML`) utilizando `.html()`
- Para modificarlo `.html('nuevo contenido HTML')`
- Podemos utilizar también `.text()` o `.text('nuevo texto')`
- **Ejercicio ¿Cuál es la diferencia entre los dos?**

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - **Valores**
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Obteniendo y cambiando el valor de un elemento HTML

- Para interactuar con los valores un los elementos `input`, podemos utilizar:

```
// Seteamos
$('#password').val('mypass123')

// Obtenemos
var user = $('#username').val()
```

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - **Atributos**
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Obteniendo y cambiando el atributo de un elemento HTML

- Podemos obtener o modificar cualquier atributo de un elemento HTML así:

```
//Leer
$('#link').attr('class')

//Modificar
$('#link').attr('class', 'titulo')

// Puede recibir tambien un JSON
$('#link').attr( {
  href : 'http://yahoo.com',
  title: 'Yahoo!',
  class: 'clase_css' } )
```

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - **Elementos**
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Añadiendo y eliminando elementos

Manipulando el DOM

- Tenemos: `append`, `prepend`, `after`, `before`, `remove` y `empty`

Añadiendo y eliminando elementos

Manipulando el DOM

- Tenemos: append, prepend, after, before, remove y empty
- **Ejercicio:** ¿Qué hacen cada una de las siguientes instrucciones?

```
<script>
  $('a').prepend('Link: ')
  $(" [href^='http'] ").append(" <img src='link.png'>")
  $('code').before('<hr>').after('<hr>')
  $('#a').click(function() { $('#ball').remove() } )
  $('#b').click(function() { $('#quote').empty() } )
</script>
```

Ancho y Alto de elementos

```
// valor numerico
width = $('#elem').width()

// valor CSS
width = $('#elem').css('width')

// Ancho de la ventana
width = $(window).width()
width = $(document).width()

// Seteando el ancho y Alto
$('#.box').width(100).height(100)
```

Ejercicio Escribir código que tenga un <div> de 300x50 px con un fondo #DDDDDD y 4 botones: “Obtener ancho de ventana”, “Obtener ancho de documento”, “Obtener ancho del div”, “Setear al ancho del div a 200px”

Ejercicio

Consultar en <http://api.jquery.com/> que hacen las funciones `.innerWidth()`, `.outerWidth()`, `.innerHeight()` y `.outerHeight()`.
Explicar y presentar un ejemplo funcional.

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Padres

Atravesando el DOM

```
// Seleccionar el padre directo
my_parent = $('#elem').parent()
my_parent = $('li').parent()
$('li').parent().css('font-weight', 'bold')
$('li').parent('.memo').css('list-style-type', 'circle')

// Seleccionar todos los ancestros
$('#elem').parents('div').css('background', 'yellow')

// Todos los ancestros hasta ?
$('#elem').parentsUntil('div').css('background', 'yellow')
```


Hijos

Atravesando el DOM

```
// Obtener los hijos directos de un elemento (1 nivel)
my_children = $('#elem').children()
li_children = $('#elem').children('li')

// Buscar hijos e todos lo niveles
// Necesita el parametro
li_descendants = $('#elem').find('li')

// Para seleccionar todos los descendientes
all_descendants = $('#elem').find('*')
```

Hermanos

Atravesando el DOM

```
<ul>
  <li>Item 1</li>
  <li id='two'>Item 2</li>
  <li>Item 3</li>
</ul>
```

```
// Todos los hermanos del li#two
$('#two').siblings().css('font-weight', 'bold')

// Solo los que tienen clase .new
$('#two').siblings('.new').css('font-weight', 'bold')
```

Ejercicio siblings excluye el elemento que lo llama. ¿Cómo deberíamos hacer si necesitamos que este también sea incluido?

Ejercicio siblings excluye el elemento que lo llama. ¿Cómo deberíamos hacer si necesitamos que este también sea incluido?

```
$( '#two' ).parent().children( '.new' ).css( 'font-weight', 'bold' )

// 0 usar .andSelf()
$( '#two' ).siblings( '.new' ).andSelf().css( 'font-weight', 'bold' )
```

Muchos formas más de atravesar el DOM

```
$('#ul>li').first().css('text-decoration', 'underline')  
  
$('#ul>li').last().css('font-style', 'italic')  
  
$('#ul>li').eq(1).css('font-weight', 'bold')  
  
$('#ul>li').filter(':even').css('background', 'cyan')  
  
$('#ul>li').not('#new').css('color', 'blue')  
  
$('#ul>li').has('ol').css('text-decoration', 'line-through')
```

Función is()

Devuelve true o false si el elemento cumple con la condición que le especificamos:

```
<script>
$( 'button' ).click(function() {
    var elem = ''
    if ( $(this).parent().is('div')) elem = 'div'
    else elem = 'span'
    $( '#info' ).html( 'You clicked a ' + elem )
})
</script>
```

Función \$.each()

```
<script>
  pets = {
    Scratchy : 'Guinea Pig',
    Squeeky : 'Guinea Pig',
    Fluffy : 'Rabbit',
    Thumper : 'Rabbit',
    Snoopy : 'Dog',
    Tiddles : 'Cat'
  }

  guineapigs = []

  $.each(pets, function(name, type){
    if (type == 'Guinea Pig') guineapigs.push(name)
  })

  $('#info').html('The guinea pig names are: ' + guineapigs.
    join(' & '))
</script>
```

La función \$.map()

Basado en el ejemplo anterior:

```
guineapigs = $.map(pets, function(type, name) {  
  if (type == 'Guinea Pig') return name  
})
```

Retorna todos los valores de nuestra función anónima en un arreglo. Esto nos evita tener que crear y popular el arreglo.

Tomar en cuenta que el orden de los parámetros es diferente entre \$.each() y \$.map()

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Existen varios métodos para manejar eventos utilizando jQuery:

```
// Adjuntar evento directo a boton (metodo 'click')
$( "#helloBtn" ).click(function( event ) {
    alert( "Hello." );
});

// Adjuntar evento directo a boton (metodo 'bind')
$( "#helloBtn" ).bind( "click", function( event ) {
    alert( "Hello." );
});

// Desde jQuery 1.7, adjuntar evento directo a boton (metodo 'on').
$( "#helloBtn" ).on( "click", function( event ) {
    alert( "Hello." );
});

// Desde jQuery 1.7, Adjuntar evento al body que escuchara por clics en cualquier boton
// de la pagina
$( "body" ).on({
    click: function( event ) {
        alert( "Hello." );
    }
}, "button" );

// Lo mismo que la anterior con diferente sintaxis
$( "body" ).on( "click", "button", function( event ) {
    alert( "Hello." );
});
```

- El método más recomendado desde la versión 1.7 es `.on()`

- El método más recomendado desde la versión 1.7 es `.on()`
- Detalles y ventajas en <http://learn.jquery.com/events/handling-events/>

```
// Un evento un handler
$( "p" ).on( "click", function() {
    console.log( "<p> was clicked" );
});

// Multiple eventos un mismo handler
$( "div" ).on( "mouseenter mouseleave", function() {
    console.log( "mouse hovered over or left a div" );
})

// Múltiples eventos múltiples handlers
$( "div" ).on({
    mouseenter: function() {
        console.log( "hovered over a div" );
    },
    mouseleave: function() {
        console.log( "mouse left a div" );
    },
    click: function() {
        console.log( "clicked on a div" );
    }
});
```

Desconectando eventos de los elementos

```
// Unbinding a particular click handler, using a reference
// to the function
var foo = function() {
    console.log( "foo" );
};

var bar = function() {
    console.log( "bar" );
};

$( "p" ).on( "click", foo ).on( "click", bar );

// foo will stay bound to the click event
$( "p" ).off( "click", bar );
```

Algunos eventos posibles

- **Navegador:** `.resize()`, `.scroll()`

Algunos eventos posibles

- **Navegador:** `.resize()`, `.scroll()`
- **Formulario:** `.blur()`, `.change()`, `.focus()`, `.focusin()`, `.focusout()`, `.select()`, `.submit()`

Algunos eventos posibles

- **Navegador:** `.resize()`, `.scroll()`
- **Formulario:** `.blur()`, `.change()`, `.focus()`, `.focusin()`, `.focusout()`, `.select()`, `.submit()`
- **Teclado:** `.keydown()`, `.keypress()`, `.keyup()`

Algunos eventos posibles

- **Navegador:** `.resize()`, `.scroll()`
- **Formulario:** `.blur()`, `.change()`, `.focus()`, `.focusin()`, `.focusout()`, `.select()`, `.submit()`
- **Teclado:** `.keydown()`, `.keypress()`, `.keyup()`
- **Mouse:** `.click()`, `.dblclick()`, `.hover()`, `.mousedown()`, `.mouseenter()`, `.mouseleave()`, `.mousemove()`, `.mouseout()`, `.mouseover()`, `.mouseup()`, `.toggle()`

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

```
// Ocultar todos los elementos p
$( "p" ).hide();

// Mostrar todos los div con clase hide
$( "div.hidden" ).show(); // 'fast' , 'slow', 'normal',
    '1000'

// ANIMACIONES
// Ocultar todos los p con una animacion slide up con
    duracion de 0.8 segundos
$( "p" ).slideUp( 800 );

// Mostrar todos los div.hidden con una animacion slideDown
    de 0.6 segundos
$( "div.hidden" ).slideDown( 600 );

// Alternar segun el estado actual
$( "p" ).toggle(); // 'slow', '1800'. .slideToggle() .
    fadeToggle()
```

Hacer algo después de la animación

MAL

```
// La siguiente funcion de cadena no espera a que finalice  
    la animacion  
$( "p.hidden" ).fadeIn( 750 ).addClass( "lookAtMe" );
```

BIEN

```
// Se debe utilizar una funcion callback  
$( "p.hidden" ).fadeIn( 750, function() {  
    // this = elemento DOM sobre el cual se acaba de aplicar  
    la animacion  
    $( this ).addClass( "lookAtMe" );  
});
```

Más info

<http://api.jquery.com/category/effects/>

<http://try.jquery.com/>

Contenido I

- 1 Recomendaciones
- 2 Introducción
- 3 Incluyendo jQuery
- 4 Document Ready
- 5 Sintaxis
- 6 El DOM
 - Contenido
 - Valores
 - Atributos
 - Elementos
 - Atravesando el DOM
- 7 Eventos
- 8 Efectos
- 9 Ejercicio

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

Lista 2:

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla

Lista 2:

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla
- Jamón

Lista 2:

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla
- Jamón
- Queso

Lista 2:

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla
- Jamón
- Queso

Lista 2:

- Coca Cola

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla
- Jamón
- Queso

Lista 2:

- Coca Cola
- Leche

Ejercicio (Entrega 5 de enero)

- Crear un documento HTML con un encabezado, dos párrafos y un botón. Al pulsar el botón se debe ocultar todo el HTML.
- Basado en el anterior. Ahora al pulsar el botón debe ocultarse dicho botón.
- Basado en el ejercicio anterior. Suponemos que el encabezado y el primer párrafo tienen el atributo `class="intro"`. Deberás ocultar dicho párrafo al darle clic al botón.
- Seleccionar el primer elemento de la primera lista y ocultarlo al darle clic a este elemento.

Lista 1:

- Tortilla
- Jamón
- Queso

Lista 2:

- Coca Cola
- Leche
- Té

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.
- Crear un documento con una tabla. Poner el fondo de las filas pares en rojo. A continuación poner el fondo de las filas impares en verde.

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.
- Crear un documento con una tabla. Poner el fondo de las filas pares en rojo. A continuación poner el fondo de las filas impares en verde.
- Utilizar el evento hover para mostrar un mensaje en un `d#mensaje` cuando nos posicionamos sobre un el y otro cuando salgamos de el.

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.
- Crear un documento con una tabla. Poner el fondo de las filas pares en rojo. A continuación poner el fondo de las filas impares en verde.
- Utilizar el evento hover para mostrar un mensaje en un `div` cuando nos posicionamos sobre un `td` y otro cuando salgamos de él.
- Utilizar los métodos `focus()` y `blur()` para cambiar el color de dos cuadros de texto cuando posicionamos el foco y cuando lo retiramos.

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.
- Crear un documento con una tabla. Poner el fondo de las filas pares en rojo. A continuación poner el fondo de las filas impares en verde.
- Utilizar el evento hover para mostrar un mensaje en un `div` cuando nos posicionamos sobre un `div` y otro cuando salgamos de él.
- Utilizar los métodos `focus()` y `blur()` para cambiar el color de dos cuadros de texto cuando posicionamos el foco y cuando lo retiramos.
- Crear un documento que presente un formulario con un cuadro de texto “Intereses” y dos botones: “agregar” y “eliminar”. Al dar clic en el botón “agregar” se debe agregar un cuadro de texto al formulario. Al dar clic en el botón “eliminar” se debe eliminar el último cuadro de texto.

Ejercicio II

- Ahora ocultar los primeros elementos de las dos listas.
- Crear un documento con una tabla. Poner el fondo de las filas pares en rojo. A continuación poner el fondo de las filas impares en verde.
- Utilizar el evento hover para mostrar un mensaje en un `d#mensaje` cuando nos posicionamos sobre un `el` y otro cuando salgamos de `el`.
- Utilizar los métodos `focus()` y `blur()` para cambiar el color de dos cuadros de texto cuando posicionamos el foco y cuando lo retiramos.
- Crear un documento que presente un formulario con un cuadro de texto “Intereses” y dos botones: “agregar” y “eliminar”. Al dar clic en el botón “agregar” se debe agregar un cuadro de texto al formulario. Al dar clic en el botón “eliminar” se debe eliminar el último cuadro de texto.
- Agregar un botón submit “guardar intereses” al formulario anterior que permita enviar el formulario por POST y escribir código PHP que permita guardar todos los cuadros de texto en una BDD.

Ejercicio

- Escribir código PHP que permita recuperar de la BDD todos los intereses guardados en el ejercicio anterior y los cargue en un formulario para poder modificarlos. La modificación implica cambiar, agregar o eliminar intereses. Se deben generar tantos cuadros de texto como intereses se hayan guardado.