

Índice

Introducción	3
Guía rápida JQuery	4
Selectores	4
Selectores propios de JQuery	4
Refinar selecciones (filtros)	5
Recorrer selecciones.....	6
Obtener y modificar atributos.....	6
Obtener y modificar estilos	6
Eventos más comunes	7
Efectos	7
Ocultar y mostrar.....	7
Fade	7
Slide	7
Animaciones	7
Dimensiones	8
Posiciones	8
Lo que debemos dominar bien de JQuery.....	9
Ejercicios propuestos.....	11
Referencias	14

Introducción

Recientemente he terminado de impartir el curso de certificación de profesionalidad "Programación con lenguaje orientado a objetos y bases de datos relacionales" (IFCD0112). Ha sido una experiencia muy satisfactoria. Después de 6 meses los alumnos han salido bien preparados y con muchas ganas a hacer sus prácticas profesionales 😊

Me ha quedado mucho material que hice para el curso, y que seguramente te será de mucha utilidad si deseas aprender a programar, y también si ya tienes experiencia pero necesitas aprender rápidamente alguna tecnología con la que no habías trabajado antes.

Para poner estos materiales al alcance de todos, los he comenzado a publicar en [mi blog](#). En el primer artículo de esta serie: [Jquery para los que conocen Javascript](#), explico los elementos fundamentales de esta librería mediante ejemplos.

Este pdf es un regalo para los programadores que hayan leído el artículo. Tiene una guía rápida que te puede ayudar a encontrar rápidamente lo que necesitas cuando estés practicando, ejercicios propuestos y un resumen de lo que debes dominar. Con el apoyo de este material **tú mismo puedes orientar tu aprendizaje y comprobar si ya dominas esta útil herramienta.**

Guía rápida JQuery

Selectores

- jQuery permite obtener un conjunto de elementos html que casen con un selector css.
- Permite todos los selectores que conocemos en css3.
- Con el alias \$. Por ejemplo:

<code>\$("*")</code>	– todos los elementos en el documento
<code>\$("a")</code>	– todos los a
<code>\$("#mielemento")</code>	– un elemento con id mielemento
<code>\$("p a")</code>	– todos a hijos de p
<code>\$("li.nombreClase")</code>	– los elementos li de la clase nombreClase
<code>\$("h1+p")</code>	– todos los p inmediatamente precedidos de h1
<code>\$("img[alt]")</code>	– todos los img con atributo alt
<code>\$("button[id*=boton]")</code>	– todos los button con id que contenga la cadena boton

- Para utilizar metacaracteres (tales como `!"#$%&'()*+,-./:;<=>?@[\\]^`{|}~`), hay que poner dos barras invertidas: `\\`.

Ej. Para un elemento con `id="foo.bar"`, se usa `$("#foo\\.bar")`

Selectores propios de JQuery

Obtener los elementos con ciertas condiciones en un conjunto seleccionado:

<code>:button</code>	– Cualquier input del tipo botón (button, submit y reset).
<code>:checkbox</code>	– Objetos input del tipo checkbox(input[type=checkbox]).
<code>:checked</code>	– Todos los objetos checkbox o radio button marcados.
<code>:disabled</code>	– Todos los objetos deshabilitados.
<code>:enabled</code>	– Todos los objetos habilitados.
<code>:file</code>	– Todos los objetos input de tipo file.
<code>:focus</code>	– Objetos que tienen el foco.
<code>:header</code>	– Todos los elementos del tipo header (h1, h2, h3, h4,h5, h6).
<code>:hidden</code>	– Objetos que están ocultos.
<code>:image</code>	– Todos los objetos input del tipo image(input[type=image]).
<code>:input</code>	– Objetos del formulario (input, select, textarea, button).
<code>:password</code>	– Todos los objetos input de tipo password.
<code>:radio</code>	– Todos los objetos input de tipo radio

:reset	– Todos los objetos input de tipo reset.
:selected	– Objetos del tipo option que están seleccionados.
:submit	– Todos los objetos input de tipo submit.
:text	– Solo elementos input del tipo texto (input[type=text]).

Obtener los elementos en cierta posición en un conjunto seleccionado:

:eq([index -index])	– Objeto en la posición index en el conjunto seleccionado.
:even	– Los objetos en las posiciones pares en el conjunto seleccionado.
:first-child	– Todos primeros hijos de supadre.
:first-of-type	– Todos los primeros elementos entre los hermanos del mismo tipo.
:first	– El primero de la selección.
:gt(index)	– Los elementos en una posición superior a index.
:last-child	– Todos últimos hijos de supadre.
:last-of-type	– Todos los últimos elementos entre los hermanos del mismo tipo.
:last	– El último de la selección.
:lt(index)	– Los elementos en una posición inferior a index.
:nth-child()	– Todos los elementos que son el enésimo hijo de su padre
:nth-last-child()	– Todos los elementos que son el enésimo hijo de su padre, contando desde el final
:nth-last-of-type()	– Todos los elementos que son el enésimo de su tipo en su padre
:nth-of-type ()	– Todos los elementos que son el enésimo de su tipo en su padre, desde el final
:odd	– Los objetos en las posiciones impares en el conjunto seleccionado.
:only-child	– Todos los elementos que son el único hijo de su padre.
:only-of-type	– Todos los elementos que no tienen hermanos del mismo tipo.
:parent	– Todos los elementos que tienen al menos un hijo.

Refinar selecciones (filtros)

\$('#ul li').eq(5)	– el sexto elemento
\$('#ul li').filter('.current')	– los elementos que posee la clase 'current'
\$('#ul li').first()	– el primer elemento

<code>\$('div.foo').has('p')</code>	– el elemento div.foo contiene elementos <p>
<code>\$('input').is(':focus')</code>	– el elemento input que tiene el foco
<code>\$('ul li').last()</code>	– el último elemento
<code>\$(":checkbox") .map(function() { return this.id; })</code>	– retorna un objeto JQuery con los id de todos los checkbox
<code>\$('h1').not('.bar')</code>	– el elemento h1 no posee la clase 'bar'
<code>\$('li').slice(2, 4)</code>	– los elementos li tercero y cuarto (posiciones 2 y 3)

Recorrer selecciones

```
$(p).each( function (i) {
    if (i%2==0) {
        $(this).css('background-color', '#eee');
    } else {
        $(this).css('background-color', '#ccc');
    }
})
```

Obtener y modificar atributos

<code>.attr()</code>	Toma el valor de un atributo para el primer elemento en el conjunto de elementos seleccionados, o da valor a uno o más atributos a cada uno de los elementos seleccionados.
<code>.removeAttr()</code>	Elimina un atributo de cada uno de los elementos seleccionados.
<code>.html()</code>	Toma el contenido html del primer elemento en el conjunto de elementos seleccionados, o cambia el contenido html a uno o más elementos para cada uno de los elementos seleccionados.
<code>.val()</code>	Toma el valor actual del primer elemento en el conjunto de elementos seleccionados, o da valor a cada uno de los elementos seleccionados

Obtener y modificar estilos

<code>.addClass()</code>	Adiciona la clase o clases a cada elemento en el conjunto de elementos seleccionado.
<code>.hasClass()</code>	Determina si alguno de los elementos seleccionados tiene la clase indicada.
<code>.removeClass()</code>	Elimina una clase, múltiples clases, o todas las clases de cada uno de los elementos seleccionados.

`.toggleClass()` Adiciona o elimina una o más clases de cada elemento en el conjunto de elementos seleccionados, dependiendo de la presencia de las clases o del valor del argumento de estado.

Eventos más comunes

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave			
mousedown			
mouseup			

Efectos

Ocultar y mostrar

`show()` Muestra.
`hide()` Oculta.
`toggle()` Oculta lo visible y muestra lo oculto.

Fade

`fadeIn()` Muestra con efecto fade un elemento oculto.
`fadeOut()` Oculta con efecto fade un elemento visible.
`fadeToggle()` Oculta con efecto fade lo visible y muestra lo oculto con efecto fade.
`fadeTo()` Realiza el fade hasta una opacidad indicada (entre 0 y 1).

Slide

`slideDown()` Bajar un elemento deslizándose.
`slideUp()` Subir un elemento deslizándose.
`slideToggle()` Si el elemento se había deslizado hacia arriba lo desliza hacia abajo y viceversa.

Animaciones

`animate()` Crear animaciones personalizadas
`stop()` Detiene una animación antes que haya terminado.

Dimensiones

`width()`

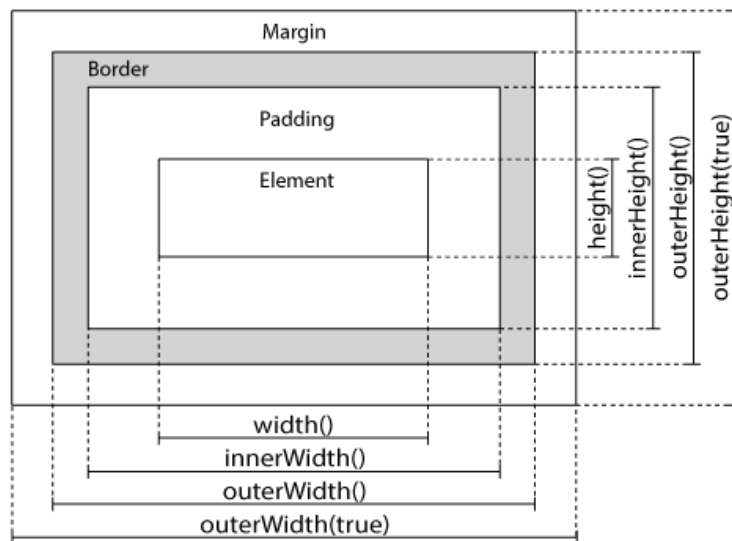
`height()`

`innerWidth()`

`innerHeight()`

`outerWidth()`

`outerHeight()`



Posiciones

`offset()` Fija o retorna las coordenadas de los elementos seleccionados en relación a la página (offset)

`offsetParent()` Retorna el primer elemento padre que tenga una posición que no sea estática (fixed, relative o absolute)

`position()` La posición de un elemento relativa a su elemento padre.

`scrollLeft()` Fija o retorna la posición de la barra de scroll horizontal de los elementos seleccionados

`scrollTop()` Fija o retorna la posición de la barra de scroll vertical de los elementos seleccionados

Lo que debemos dominar bien de JQuery

1. Utilizar JQuery en las funciones para validar formularios y manejar eventos:

- Acceder y manipular los elementos del DOM:
 - Obtener elementos html:
 - `$("<selector>")`
 - Selectores css3.
 - Selectores propios de JQuery.
 - Las selecciones se pueden refinar (has, not, filter, first, last, eq).
 - Comprobar selecciones: `if ($('<selector>').length) { ... }`
 - Obtener valores:
 - select simple, textarea, input text – `$(' <selector> ').val()`
 - select múltiple – `$(' <selector> ').val()` (devuelve un arreglo)
 - radio – `$(' <selector>:checked').val();`
 - checkbox – hay que obtener todos los valores:

```
var checkboxValues = new Array();
$('input[name="seleccion[]"]:checked').each(
    function() { checkboxValues.push( $(this).val() ); });
```
 - Modificar valores:
 - select simple, textarea, input text – `$(' <selector> ').val(<valor simple>)`
 - select múltiple – `$(' <selector> ').val(<arreglo>)`
 - radio, checkbox – `$(' <selector> ').eq(<posición>).prop(' checked', <valor>);`
 - Obtener y modificar atributos:
 - `$(<selector>).attr("<atributo>" [, "valor"])`
 - Obtener y modificar estilos:
 - `$(<selector>).css("<propiedad css>" [, "valor"])`
 - `$(<selector>).hasClass("<nombre clase>")`
 - `$(<selector>).removeClass("<nombre clase>");`
 - `$(<selector>).addClass("<nombre clase>")`
 - Obtener y modificar contenido:
 - `$(<selector>).html(["<código html>"])`
 - Recorrer una selección (each).

2. Manejar eventos:

- Programar el manejador de evento:
 - Cuando la página esté lista: `$(document).ready(function(){ ... }) ;`
- Obtener el evento lanzado desde la función que maneja el evento:
 - en la función que maneja el evento: `$("a").click(function(elEvento) { ... })`
- Obtener el elemento HTML que lanzó el evento, en la función que maneja el evento:
 - `$(this)`
 - `$(elEvento.target)`
- Obtener información del evento lanzado, en la función que maneja el evento:
 - el elemento HTML que lanzó el evento: `$(elEvento.target)`.
 - en los eventos de ratón, posición del puntero en la ventana del navegador:

`elEvento.pageX`

`elEvento.pageY`

- en los eventos del teclado (keydown, keypress y keyup):

`elEvento.keyCode` – código de la tecla pulsada.

`elEvento.charCode` – código del carácter asociado a la tecla.

- Cancelar el evento lanzado, en la función que maneja el evento:
 - en la función que maneja el evento: `elEvento.preventDefault();`

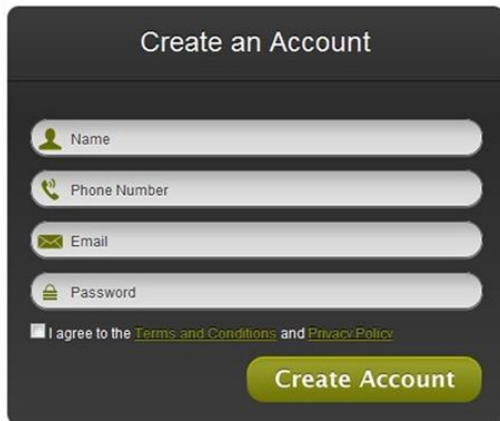
3. Efectos:

- Ocultar y mostrar (show, hide y toggle).
- Fade (fadeIn, fadeOut, fadeTo, fadeToggle).
- Slide (slideDown, slideUp, slideToggle).
- Animaciones (animate, stop).

Ejercicios propuestos

Ejercicio 1

Crear el siguiente formulario y validarlo utilizando JQuery. Solo es opcional el teléfono. Comprobar que el valor de todos los campos cumple ciertas reglas sintácticas. Solicitar confirmación cuando el usuario envía el formulario.



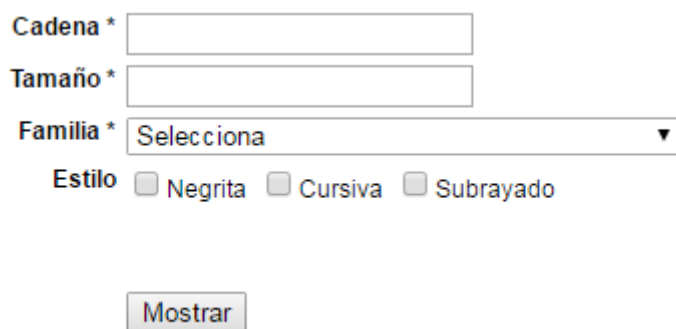
A dark-themed 'Create an Account' form. It features four input fields: 'Name' (with a person icon), 'Phone Number' (with a phone icon), 'Email' (with an envelope icon), and 'Password' (with a lock icon). Below the fields is a checkbox labeled 'I agree to the [Terms and Conditions](#) and [Privacy Policy](#)'. At the bottom is a green 'Create Account' button.

Ejercicio 2

Crear una página web para mostrar un texto introducido por el usuario en el formato que el propio usuario haya indicado. Para ello se creará un formulario para introducir: una cadena, el tamaño de letra, una familia tipográfica e indicar si utiliza negrita, cursiva o subrayado.

Excepto los estilos los demás campos son obligatorios. Validar que el tamaño de letra sea un número.

Utilizar Javascript para las validaciones.



A light-themed form for text formatting. It contains four fields: 'Cadena *' (text input), 'Tamaño *' (text input), 'Familia *' (dropdown menu with 'Selecciona' selected), and 'Estilo' (checkboxes for 'Negrita', 'Cursiva', and 'Subrayado'). A 'Mostrar' button is at the bottom.

Ejercicio 3

Realizar una página que muestre el valor de la letra de un número de identificación fiscal (NIF) a partir de los números introducidos por el usuario.

Mostrar el mensaje de error y la letra en el mismo html, utilizando un estilo diferente en cada caso (ver imagen ilustrativa).

El algoritmo es el siguiente para determinar la letra es el siguiente:

- Comprobar que el número está entre 0 y 999999999
- Calcular el resto de la división entera del número de DNI y el número 23
- Seleccionar la letra dentro del arreglo de letras siguiente:
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];
Esto es, si el resto de la división es 0, la letra del DNI es la T, si el resto es 1 la letra es la R, y así sucesivamente.

Número *	<input type="text"/>	Número *	<input type="text" value="12345678"/>
Letra	ERROR: Introduce un número	Letra	Z
	<input type="button" value="Calcular"/>		<input type="button" value="Calcular"/>

Ejercicio 4

Permitir que el usuario introduzca hasta 10 números. Emitir mensaje de error si alguno de los valores introducidos no es un número o si no se ha introducido ningún valor. A partir de la lista introducida (ignorar los campos vacíos), determinar:

- La cantidad de números mayores que 10.
- La sumatoria de todos los números.
- El promedio de todos los números.

Mostrar los resultados y los mensajes de error en la propia página.

Ejercicio 5

Permitir que el usuario seleccione una provincia e introduzca un código postal, para indicarle si el código postal introducido pertenece a la provincia seleccionada o no.

Consultar códigos postales por provincias en <http://www.codigopostal.org/provincias>.

Tener en cuenta que, ambos campos son obligatorios y que el código postal tenga el formato adecuado.

Mostrar los resultados y los mensajes de error en la propia página.

Ejercicio 6

Crear una página llamada “Reservar mesa” que incluye un formulario que permite indicar:

- El día que se desea reservar en un calendario (obligatorio).
- Cantidad de personas (obligatorio).
- Indicar si se desea reservar el día entero o una franja horaria. Por defecto está seleccionado reservar en una franja horaria.
- En el caso que se quiera reservar en una franja horaria, se muestran los campos para seleccionar la hora inicial y final (obligatorios).

- Indicar si alguno de los comensales es intolerante al gluten o si es alérgico a algunos alimentos.
- En el caso que alguien sea alérgico, se muestra una lista de alérgenos para seleccionar a cuáles (obligatorio).
- Opcionalmente se puede introducir una nota.

Además tener en consideración:

- Distribuir los campos de manera que sea cómodo para el usuario.
- Emitir los mensaje de error y de información en el mismo html.
- Destacar los campos con errores.

Referencias

<http://www.w3schools.com/jquery>

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

http://www.w3schools.com/Css/css_pseudo_classes.asp

http://www.w3schools.com/Css/css_pseudo_elements.asp