



DECODED

ACADEMY

FULL STACK DEVELOPER BOOTCAMP

Desarrollamos
{ talento }

jQuery

Intro

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por [John Resig](#), que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

Documentación oficial

- [Página principal](#)
- [Documentación del API](#)
- [Descarga](#)

Instalación

Cargando la referencia directamente desde un CDN

```
<script type ="text/javascript"  
src="//cdnjs.cloudflare.com/ajax/libs/jquery/1.8.2/jquery.js"> </script>
```

Descargando en local y cargando el mismo desde la página.

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

Uso

Iniciando jQuery

Podemos iniciar lo directamente en la página.

```
<script type="text/javascript">
$(document).on('DOMContentLoaded', function() {
    console.log('hola mundo, estoy usando jQuery');
});
</script>
```

Iniciando Jquery

O cargarlo desde un js externo.

```
<!-- index.html -->
<script src="main.js"></script>

// main.js
$(document).ready(function() {
  console.log('hola mundo, estoy usando jQuery');
});
```

Seleccionando nodos del DOM

Se puede seleccionar un nodo o varios mediante selectores estilo CSS

```
var nodeTitle = $('section h1.title');
```

Mediante la clase del elemento.

```
var elemento = $('.contenedor');  
<div class="contenedor">Contenedor de clase</div>
```

Mediante el id del elemento.

```
var miElemento = $('#contenedor');

miElemento.on('click', function() {
  console.log('Contenedor pulsado');
}) ;

<div id="contenedor">Contenedor</div>
```

Mediante la clase del elemento.

```
var misElementos = $('.contenedor');
misElementos[0].on('click', function() {
  console.log('Contenedor pulsado');
});

<div class="contenedor">Contenedor de clase</div>
```

Mediante el tipo de la etiqueta.

```
var misElementos = $('div[name="contenedor"]');
misElementos[0].on('click', function() {
    console.log('Contenedor pulsado');
});
<div name="contenedor">Contenedor</div>
```

Manipulación del DOM

.addClass()

Agrega las clases especificadas a cada elemento en el conjunto de elementos coincidentes.

```
$( '#miElemento' ).addClass( 'soyUnaClase' );
```

.empty()

Elimina todos los nodos hijos del conjunto de elementos coincidentes del DOM.

.detach()

Elimine el conjunto de elementos coincidentes del DOM.

.html()

Obtiene los contenidos HTML del primer elemento en el conjunto de elementos coincidentes o establece los contenidos HTML de cada elemento coincidente.

.append()

Inserte el contenido, especificado por el parámetro, al final de cada elemento en el conjunto de elementos coincidentes.

.prepend()

Inserte el contenido, especificado por el parámetro, al comienzo de cada elemento en el conjunto de elementos coincidentes.

Más info

- [Docs](#)

Creación de un nodo

jQuery puede recibir varios tipos de strings, si es un string tipo selector CSS `$('.item')` buscará ese elemento o elemtos en el DOM y devolverá la instancia del node para que podamos manipularlo.

Pero si recibe un string tipo HTML `$('<div class="item">Hola mundo</div>')`; creará un nodo nuevo y devolverá su instacia para que podamos insertarlo en el DOM o hacer los que queramos con él.

Creación de elementos e inserción

```
var miNuevoElemento = $('

Hola mundo</div>');  
$('.container').append(miNuevoElemento);


```

Eventos

Evento: Algo que ocurre

Manejador de evento: Una función que contiene el código a ejecutar cuando se produce un evento

Objeto evento: Un objeto que se genera automáticamente con cada evento y contiene info y funcionalidades asociadas al evento.

on()

Activa manejadores de evento.

Asociar un comportamiento ante cualquier tipo de evento, por medio de una función manejadora de evento.

```
$('selector').on('click', function(objetoEvento) {  
    //Aquí se introduce toda la funcionalidad.  
});
```

Tipos de Eventos

- Docs on()
- Eventos nativos

off()

Desactiva manejadores de evento.

- Una función manejadora en concreto.
- Todos los manejadores de un tipo de evento.
- Todos los manejadores de todos los eventos.

one()

Como on(), pero el evento asociado sólo se ejecutará una vez. Sería lo mismo que hacer un .on() y un .off() en su primera llamada.

Objeto event

El objeto evento de Javascript normalizado y extendido por JQuery

- Datos del evento.
- Funcionalidades asociadas.

event.type

Te dice el tipo de evento que se está produciendo.

Para probarlo, aprendemos a definir un manejador en distintos tipos de eventos a la vez.

```
$('a').on('click', function(e) {  
    console.log(e.type); // click  
});
```

event.preventDefault()

Previenes el comportamiento por defecto que el navegador realiza predeterminadamente cuando se produce un evento.

- Se propaga el evento

ejemplo PreventDefault

```
$('a').on('click', function(e) {  
    e.preventDefault();  
    console.log('Ha pulsado el enlace. Pero no irá al link pulsado');  
});  
Enlace
```

this

Hace referencia al closure donde está situado el mismo.

```
$("a").on("click", function(e) {  
    e.preventDefault();  
    console.log($(this).text());  
    console.log($(this).attr("href"));  
});
```

evento.target

La propiedad evento.target también hace referencia al nodo donde se inició en el evento:

```
$("a").on("click", function(e) {  
    e.preventDefault();  
    console.log($(e.target).text());  
    console.log($(e.target).attr("href"));  
});
```

Otro eventos que se pueden añadir.

- click() igual que: on('click',fun)
- ready() igual que: on('ready',fun)
- keypress() igual que: on('keypress',fun)
- mouseenter() igual que: on('mouseenter',fun)

Eventos de ratón

mousemove()

Tipo de evento que se ejecuta cuando mueves el ratón por encima de un elemento sobre el que has definido el evento.

e.which

Puedo saber el botón del ratón que se ha pulsado.

1. Botón izq.
2. Botón medio
3. Botón der.

Ojo: Un evento click sólo se ejecuta con el click del botón izquierdo. Para detectar cualquier tipo de pulsación puedes usar el evento mousedown.

Eventos del teclado

keydown keypress keyup

Eventos de ratón que puedes usar para asignar comportamientos.

e.which

Podemos acceder al código de la letra que se ha pulsado en un evento de teclado.

Es una información disponible en el objeto evento, al pulsar una tecla se producen los tres eventos en el orden: Keydown, Keypress, Keyup.

```
String.fromCharCode(e.which); //convierte el código de la tecla, Unicode , al carácter asociado.
```

e.shiftkey e.ctrlkey

Para averiguar si tienes pulsada alguna tecla especial en el momento en el que se produjo el evento.

```
if(e.shiftkey) {  
    console.log("shift pulsado");  
}
```

Efectos de Jquery

Funciones para realizar efectos.

- hide()
- show()
- fadeIn() fadeOut() fadeTo()
- slideUp() slideDown()
- toggle() slideToggle()

Funciones callback

Las funciones de efectos tienen la posibilidad de especificar una función con código que se ejecutará cuando terminen los efectos.

```
$('#elem').hide(2000, function() {  
  console.log('Fin!!!');  
});
```

Más información

- Effects

Animaciones personalizadas

Una función de animación perfectamente personalizable en el que usar cualquier propiedad numérica de CSS.

```
$('#elem').animate(  
{  
  'font-size': '24pt',  
  width: '200px',  
},  
2000  
) ;
```