

JQuery I



Índice

Fundamentos de jQuery.....	3
Introducción a jQuery.....	3
¿Qué es jQuery?.....	3
¿Por qué usar jQuery?.....	3
¿Qué ventajas tiene el uso de JQuery?.....	3
¿ Existen otras librerías similares a JQuery?.....	4
Cómo agregar jQuery a sus páginas web.....	5
1.- Descargando jQuery.....	5
2.- CDN de jQuery.....	6
Cuestiones iniciales: diferencia entre \$ y \$().....	6
Iniciar JQuery.....	7
Evitar conflictos con otras librerías.....	8
El elemento Selector.....	10
Syntaxis de jQuery.....	10
1.- Básicos.....	10
2.- De Atributos.....	11
Ejercicio guiado.....	11
Ejercicio propuesto.....	16

Fundamentos de jQuery

Introducción a jQuery

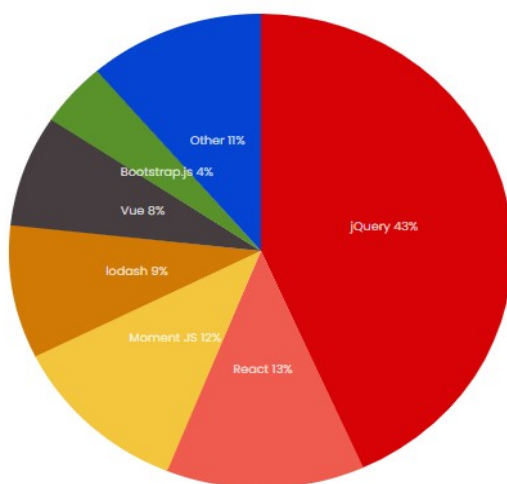
¿Qué es jQuery?

Es una biblioteca de JavaScript, diseñada por John Resig, para facilitar la **manipulación del DOM** (Document Object Model), el **manejo de eventos**, las **animaciones** y la **interacción con servidores** a través de AJAX.

Su principal ventaja es hacer que el código JavaScript sea más simple y fácil de entender, lo que permite a los desarrolladores escribir menos código para tareas típicas, como elegir y cambiar elementos de una página web. Además, jQuery proporciona una API unificada que funciona en todos los principales navegadores, lo que ayuda a resolver problemas de compatibilidad entre navegadores.

¿Por qué usar jQuery?

Hay muchas otras bibliotecas de JavaScript, pero jQuery es probablemente la más popular y también la más extensible.



Muchas de las empresas más grandes de la Web utilizan jQuery, como:

- Google
- Microsoft
- IBM
- Netflix

¿Qué ventajas tiene el uso de JQuery?

Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y utilizar AJAX de forma realmente simple.

Vemos en detalle las ventajas y desventajas:

Vetantajas	Desventajas
Es Cross-Browser: Nosotros programamos en JQuery. La librería se encargará de detectar el navegador y adaptar todo el comportamiento para que el resultado sea independiente	Rendimiento: Cargar la biblioteca de jQuery añade peso a la página, lo que puede ralentizar la carga.
Acceder y modificar elementos en un documento: JQuery dispone de un mecanismo de selección robusto y eficiente que permite acceder y modificar un documento mediante accesos rápidos a su DOM.	Redundancia de datos. Conectar una biblioteca completa por un solo efecto no es la mejor idea.
Modificar la apariencia de una página Web: podemos cambiar las clases o propiedades del CSS de los elementos.	Control limitado. Si quieres tener el máximo control de tu sitio web, sería mejor que uses JavaScript sin jQuery.
Responder a los eventos de un usuario: es capaz de interceptar una amplia variedad de eventos adaptándose al navegador.	Muchas versiones: tienes que instalar la adecuada y alojarla en tu proyecto.
Animaciones y efectos: la librería incluye una serie de animaciones y efectos del tipo (ocultar, deslizar, desvanecer, etc.).	
AJAX nativo: jQuery dispone de sus propios métodos para realizar peticiones Ajax.	
Simplificar tareas JavaScript comunes: Además de todas las características específicas de jQuery, la librería proporciona mejoras a construcciones JavaScript básicas como iteración y manipulación de tablas.	
<u>jQuery UI</u> es un conjunto selecto de interacciones, efectos, widgets y temas para la interfaz de usuario, creados a partir de la biblioteca JavaScript jQuery.	
<u>Plugins:</u> Un plugin de jQuery es simplemente un nuevo método que utilizamos para extender el objeto prototipo de jQuery. Podrías considerar cada método que viene con el núcleo de jQuery como un plugin, como <code>`get` .fadeOut()</code> o <code>`set` .addClass()</code>	

¿ Existen otras librerías similares a JQuery?

Sí, una de ellas es [Mootools](#), permite escribir código potente y flexible. Tiene APIs elegantes. Usa licencia MIT, lo que permite usar su código modificarlo. El código es fácil de leer, y permite ampliar la funcionalidad para adaptarla a sus necesidades.

Existen otros aquí te dejo una tabla con sus características:

	jQuery	jQuery UI	Dojo Toolkit	React	Zeptojs	CreateJS
Año de publicación	2006	2007	2005	2013	2010	2012
Mantenimiento	JS Foundation	JS Foundation	Dojo Foundation	Facebook	Thomas Fuchs	gskinner.com
Licencia	MIT	MIT	BSD/AFL	MIT	MIT	MIT
Colaboradores en GitHub	270 aprox.	300 aprox.	100 aprox.	1.100 aprox.	180 aprox.	40 aprox.
Particularidades	Biblioteca más popular	Foco de atención en GUI	Foco de atención en aplicaciones web	DOM virtual	Minimalista	Integración en Adobe Animate

Cómo agregar jQuery a sus páginas web

Existen varias formas de comenzar a usar jQuery en su sitio web. Puede:

- Descargar la biblioteca jQuery desde <https://jquery.com>
- Descargar a un proyecto usando npm o yarn.
- Incluir jQuery desde un CDN, como Google

1.- Descargando jQuery

Nosotros usaremos la opción de descargar la biblioteca, por evitar accesos a internet. Hay dos versiones de jQuery disponibles para descargar:

- Versión de producción: es para su sitio web en vivo. Se ha minimizado y comprimido las librerías.
- Versión de desarrollo: es para pruebas y desarrollo (código sin comprimir y legible)

Ambas versiones se pueden descargar desde jquery.com.

Ve a descargas de jquery.com y pincha en el enlace: [Descarga la versión de desarrollo sin comprimir de jQuery 3.7.1](#)

Haz click derecho sobre el código y dale a “Guardar como”. Elige la ruta de tu proyecto donde guardad el fichero.

La biblioteca jQuery es un único archivo JavaScript, y se hace referencia a él con la `<script>` etiqueta HTML (tenga en cuenta que la etiqueta `<script>` debe estar dentro de la sección `<head>`):



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Aprendiendo JQuery</title>
  <script src="jquery-3.7.1.min.js"></script>
</head>
<body>
  .....
</body>
</html>
```



Coloque el archivo descargado en el mismo directorio de las páginas donde desea utilizarlo.

2.- CDN de jQuery

Si no desea descargar y alojar jQuery usted mismo, puede incluirlo desde una CDN (red de distribución de contenido).

Google es un ejemplo de alguien que aloja jQuery:



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Aprendiendo JQuery</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
</head>
<body>
  .....
</body>
</html>
```

Cuestiones iniciales: diferencia entre \$ y \$()

Los métodos y propiedades estáticos de jQuery están en el *namespace* \$ y reciben y devuelven automáticamente la selección como this. Como por ejemplo: \$.each(), \$.ajax(), \$.trim().

Por otro lado, \$() es la selección/creación de elementos y devuelve un objeto jQuery sobre el que encadenas: .addClass(), .on(), .hide(), etc.

Por ejemplo:

```
// Seleccionar elementos
const $enlaces = $("a"); // $() devuelve un objeto jQuery
```

```
// Usar utilidades de jQuery en el "namespace" $
$.each($enlaces, function (i, enlace) {
    console.log(i, enlace);
});
```

Es una buena norma no escrita, una convención, de poner \$ delante de las variables que almacenan resultados de jQuery, simplemente para identificar que es "algo" de jquery. \$ no añade ni quita nada a la variable, así que, "\$enlaces" es el nombre de una variable y además, sabemos que es de jquery porque le pusimos el \$ delante.

Existen algunos casos en los que los métodos de objetos y los métodos del núcleo tienen el mismo nombre, como `methods` \$.each() y `methods` .each(). En estos casos, tenga mucho cuidado al leer la documentación para asegurarse de que está consultando el método correcto.

Iniciar JQuery

Una página no se puede manipular de forma segura hasta que el documento esté listo. jQuery detecta este estado automáticamente. La forma de hacerlos es con el siguiente código:



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

Existe una forma abreviada que es la siguiente:



```
$(function(){
    // jQuery code go here...
});
```

También podemos pasar una función a \$(document).ready() de la manera siguiente:



```
function init( jQuery ) {
    // jQuery code go here...
}

$( document ).ready( init );
```

Incluso podría escuchar la carga del documento en el objeto “window” con el método “load”:



```
function init( jQuery ) {  
    // jQuery code go here...  
}  
  
$( window ).on( "load", init );
```

Evitar conflictos con otras librerías

La biblioteca jQuery está dentro del *namespace* jQuery. Los objetos globales también se almacenan dentro del espacio de nombres, así que, no habría conflictos entre jQuery u otras biblioteca (como MooTools o YUI).

Hay una salvedad: si usamos otra librería de JavaScript que también quiere usar el símbolo \$: entonces se pueden pisar entre ellas. Para evitarlo, justo después de cargar jQuery podemos activarlo en “**modo sin conflictos**”, de forma que deje libre el símbolo \$ y usemos otro nombre para jQuery.

La forma de activar el “modo sin conflictos” es liberando el símbolo \$ y se hace con **jQuery.noConflict()**. Esto viene a decir: “Deja de usar el símbolo \$ y además devuelve el objeto jQuery”, dicho objeto lo recogemos en una variable nueva. Haríamos algo así: \$j = jQuery.noConflict().



```
<script src="jquery.js"></script>  
<script>  
  
var $j = jQuery.noConflict();  
  
$j(document).ready(function() {  
    $j( "div" ).hide();  
});  
</script>
```


Otra forma:



```
<script src="jquery.js"></script>
<script>

var $j = jQuery.noConflict();

function init(){
    $( "div" ).hide();
}

$(document).ready(init); // También $j(init);
</script>
```

El elemento Selector

Sintaxis de jQuery

La sintaxis jQuery está hecha a medida para seleccionar elementos HTML y realizar alguna acción en dichos elementos.

La sintaxis básica es: **\$(selector) .action ()**

- Un signo \$ para definir/acceder a jQuery
- Un (selector) para "consultar (o encontrar)" elementos HTML
- Una acción jQuery () que se realizará en el elemento(s)

Ejemplos:

- ◆ \$(this).hide() --> oculta el elemento actual.
- ◆ \$("p").hide() --> oculta todos los elementos <p>.
- ◆ \$(".test").hide() --> oculta todos los elementos con clase="test".
- ◆ \$("#test").hide() --> oculta el elemento con id="test".

El selector de elementos jQuery selecciona elementos según el nombre del elemento.

Puedes seleccionar todos los elementos <p> de una página de esta manera: \$("p") . Veamos un ejemplo:



```
$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
});
```

Cuando un usuario hace clic en un botón, todos los elementos <p> se ocultarán. Otros selectores y su sintaxis jQuery son los siguientes:

1.- Básicos

- \$("*"): Todos los elementos del documento.
- \$("#miElemento"): Elemento con el ID miElemento.
- \$(".miClase"): Todos los elementos con la clase miClase.
- \$("[miAttribute]"): Selecciona los elementos que tienen un atributo miAttribute.
- \$("[attribute='miValue']") : Selecciona los elementos que tienen un atributo con valor miValue.
- \$("p"): Todos los elementos párrafos

2.- De Atributos

- `$("[atributo]")`: Todos los elementos con el atributo especificado.
- `$("[atributo='valor']")`: Elementos con el atributo igual al valor especificado.
- `$("[atributo!='valor']")`: Elementos con el atributo diferente al valor especificado.
- `$("[atributo^='valor']")`: Elementos con el atributo que comienza con el valor especificado.
- `$("[atributo$='valor']")`: Elementos con el atributo que termina con el valor especificado.
- `$("[atributo='valor']")*`: Elementos con el atributo que contiene el valor especificado.

Ejercicio guiado

Vamos a ver ejemplos de uso de cada uno de ellos. Partiremos de la siguiente página web:



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Práctica jQuery - Gravity Falls</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    body { font-family: Arial, sans-serif; background-color: #f4f4f4; padding: 20px; }
    .misterio { color: purple; }
    .resaltar { background-color: yellow; }
    .oculto { display: none; }
    .btn { margin: 5px; padding: 8px 12px; background-color: #444; color: white; border: none; cursor: pointer; }
  </style>
</head>
<body>
  <h1>Práctica jQuery con Gravity Falls</h1>

  <button class="btn" id="btn-ejecutar">Ejecutar</button>

  <div>
    <p id="intro">Bienvenido al bosque de Gravity Falls.</p>
    <p class="misterio">Este bosque guarda muchos secretos.</p>
    <p class="misterio" data-lugar="cabaña">La cabaña del misterio es un buen lugar para comenzar.</p>
    <p class="misterio" data-lugar="bosque">Pero el bosque esconde cosas aún más oscuras.</p>
    <p data-personaje="Dipper">Dipper está investigando un fenómeno extraño.</p>
    <p data-personaje="Mabel">Mabel ha hecho nuevos amigos brillantes.</p>
    <p data-personaje="Bill" data-enemigo="true">Bill Cipher está acechando en las sombras.</p>
    
  </div>

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

Cuando se pulse el boton con id="btn-ejecutar" se ejecutaran las siguientes acciones:

1. Usa jQuery para seleccionar todos los elementos del documento y aplicarles un borde punteado gris claro.
2. Cambia el color del texto del párrafo con ID intro a verde.
3. Subraya todos los elementos que tengan la clase misterio.
4. Resalta todos los elementos que tengan el atributo data-personaje.
5. Modifica el texto del personaje con data-personaje="Dipper" para indicar que está en la biblioteca.
6. Cambia la fuente de todos los elementos <p> a Georgia.
7. Cambia el color del texto a rojo de los elementos con el atributo data-enemigo.
8. Resalta los elementos cuyo atributo data-enemigo tenga el valor "true".
9. Aplica cursiva a los elementos con data-personaje distinto de "Mabel".
10. Cambia el fondo de los elementos cuyo atributo data-lugar comience con "caba".
11. Muestra (fadeIn) los elementos cuyo atributo data-rol termine en "final".
12. Aplica un borde rojo grueso a los elementos cuyo atributo title contenga la palabra "villano".

Lo primero a hacer es capturar el evento del click del ratón:



```
<script>
$('#btn-ejecutar').click(function() {
    .....
})
</script>
```

Nota que el botón tiene un "id" y para acceder a el con jQuery lo hacíamos con "#". Así que, tenemos \$('#btn-ejecutar'). Ahora escuchamos el evento click con \$('#btn-ejecutar').click() para que ejecute una función anónima function(){ }.

Resolvamos cada uno de los ejercicios:

1. Usa jQuery para seleccionar todos los elementos del documento y aplicarles un borde punteado gris claro.

Seleccionamos todos los elementos con: \$('*')

Accedemos al css con: \$('*').css('atributo', 'valor')

Aplicamos borde gris claro punteado: \$('*').css('border', '1px dashed lightgray')



```
<script>
$('#btn-ejecutar').click(function() {
    // 1) Selecciona todos los elementos y agrega un borde
    $('*').css('border', '1px dashed lightgray');
})
</script>
```

2. Cambia el color del texto del párrafo con ID intro a verde.

Seleccionamos el id con # de la forma: \$('#intro')

Y como en el ejercicio anterior aplicamos el css solicitado: \$('#intro').css('color', 'green')



```
<script>
$('#btn-ejecutar').click(function() {
  2) Selecciona el elemento con ID intro y cambia su color
  $('#intro').css('color', 'green');
})
</script>
```

3. Subraya todos los elementos que tengan la clase misterio.

Para seleccionar una clase lo hacemos a través del elemento “punto”: \$('.misterio')

Y como en el ejercicio anterior aplicamos el css solicitado: css('text-decoration', 'underline')



```
<script>
$('#btn-ejecutar').click(function() {
  // 3) Selecciona todos los elementos con clase misterio y subráyalos
  $('.misterio').css('text-decoration', 'underline');
})
</script>
```

4. Resalta todos los elementos que tengan el atributo data-personaje.

Los atributos se selecciona mediante el elemento corchete, este entre comillas simples: \$('[data-personaje]').

Añadimos una clase mediante addClass('nombre_clase').



```
<script>
$('#btn-ejecutar').click(function() {
  // 4) Selecciona los elementos con el atributo data-personaje
  $('[data-personaje]').addClass('resaltar');
})
</script>
```

5. Modifica el texto del personaje con data-personaje="Dipper" para indicar que está en la biblioteca.

Los atributos se selecciona mediante el elemento corchete, este entre comillas simples. `$('[data-personaje]')`.

Como queremos uno en concreto debemos igualar el atributo al que buscamos: `$('[data-personaje="Dipper"]')`.



```
<script>
$('#btn-ejecutar').click(function() {
  // 5) Selecciona los elementos donde data-personaje sea 'Dipper'
  $('[data-personaje="Dipper"]').text('Dipper está en la biblioteca.');
```

6. Cambia la fuente de todos los elementos <p> a Georgia.

Los elementos html lo indicamos con la etiqueta entre comillas simples: `$('p')`.

Con lo explicado anteriormente sobre el css solicitado: `css('font-family', 'Georgia')`



```
<script>
$('#btn-ejecutar').click(function() {
  // 6) Selecciona todos los párrafos y cambia su fuente
  $('p').css('font-family', 'Georgia');
```

7. Cambia el color del texto a rojo de los elementos con el atributo data-enemigo.

¿Te atreves con este? ¡Exacto! Seleccionas el elemento con el atributo indicado: `$('[data-enemigo]')`

Con lo explicado anteriormente sobre el css solicitado: `css('color', 'red')`



```
<script>
$('#btn-ejecutar').click(function() {
  // 7) Todos los elementos con atributo data-enemigo
  $('[data-enemigo]').css('color', 'red');
```

8. Resalta los elementos cuyo atributo data-enemigo tenga el valor "true".

Este debes ya entenderlo.



```
<script>
$('#btn-ejecutar').click(function() {
  // 8) Elementos donde data-enemigo sea igual a 'true'
  $('[data-enemigo="true"]').addClass('resaltar');
})
</script>
```

9. Aplica cursiva a los elementos con data-personaje distinto de "Mabel".

Recuerda que para indicar el “distinto” se realiza con “ != “



```
<script>
$('#btn-ejecutar').click(function() {
  // 9) Elementos donde data-personaje sea distinto de 'Mabel'
  $('[data-personaje!="Mabel"]').css('font-style', 'italic');
})
</script>
```

10. Cambia el fondo de los elementos cuyo atributo data-lugar comience con "caba".

Recuerda que para indicar el comienzo por “caba” se realiza con “ ^= “



```
<script>
$('#btn-ejecutar').click(function() {
  // 10) Elementos con atributo data-lugar que comience con "caba"
  $('[data-lugar^="caba"]').css('background-color', '#def');
})
</script>
```

11. Muestra (fadeIn) los elementos cuyo atributo data-rol termine en "final".

Recuerda que para indicar el termine por “final” se realiza con “\$= “
fadeIn hace que aparezca el elemento poco a poco de la nada.



```
<script>
$('#btn-ejecutar').click(function() {
  // 11) Elementos donde el atributo data-rol termine en "final"
  $('[data-rol$="final"]').fadeIn();
})
</script>
```

12. Aplica un borde rojo grueso a los elementos cuyo atributo title contenga la palabra "villano".

Recuerda que para indicar que contenga “villano” se realiza con “*= “



```
<script>
$('#btn-ejecutar').click(function() {
  // 12) Elementos donde el atributo title contenga "villano"
  $('[title*="villano"]').css('border', '3px solid red');
})
</script>
```

Ejercicio propuesto

UD03UT01_JQ_8_ejercicios-basicos_y_atributos