

## Integrando jQuery

<b>Integrando jQuery</b>	<b>1</b>
¿Qué aprenderás?	2
Introducción	2
Integrando jQuery	3
¿Qué es jQuery?	3
Características	3
Formas de integrar jQuery	4
Descargando el archivo:	4
CDN	6
Sintaxis jQuery	7
Selectores	8
Eventos y Funciones	9
¿Qué son los eventos?	9
Sintaxis de los eventos	10
Métodos	11
Efectos	11
Manipulación de HTML y CSS	12
Ejercicio guiado: Primeros pasos con jQuery	14
Contexto	14
Resumen	18



**¡Comencemos!**

## ¿Qué aprenderás?

- Implementar adecuadamente la librería jQuery en un proyecto web, utilizando recursos locales y externos.

## Introducción

Hasta ahora hemos experimentado con JavaScript y pudimos hacer un par de cosas interesantes, como interactuar con el usuario a través de ventanas emergentes y realizar operaciones matemáticas.

Estas habilidades, son la base para lo que abordaremos en las siguientes páginas, donde nos centraremos en la librería jQuery, que viene a simplificar la tarea de programar en JavaScript y permite agregar interactividad a un sitio web, optimizando el tiempo y la legibilidad del código.

**¡Vamos con todo!**



## Integrando jQuery

Anteriormente, conocimos un poco sobre los conceptos básicos de JavaScript; desde ahora, comenzaremos a trabajar con jQuery.

### ¿Qué es jQuery?



Imagen 1. Logo jQuery.

Fuente: [jQuery Brand Guidelines](#).

Es una biblioteca de JavaScript que nos permite escribir menos código y obtener los mismos resultados que con JavaScript puro. También, hace más sencilla la captura y manipulación de los diferentes elementos de un sitio. Esto lo realiza encapsulando el código JavaScript en métodos más cortos y simples.

Además, jQuery es software libre y de código abierto, bajo una licencia MIT y GNU que permite su uso privado y comercial.

### Características

La biblioteca de jQuery tiene las siguientes características:

- Manipulación del DOM/HTML;
- Manipulación de CSS;
- Métodos de eventos para HTML;
- Efectos y animaciones;
- Entre otras.

## Formas de integrar jQuery

Para integrar jQuery a un proyecto tenemos dos opciones. Sigue los pasos que te presentamos a continuación:

### 1. Descargando el archivo:

- **Paso 1:** baja la biblioteca desde [jQuery](https://jquery.com/), para esto descarga el archivo del mismo nombre,, haciendo clic con el botón derecho sobre el link de descarga:

#### Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file saves bandwidth and improves performance in production. You can use either with a compressed file. The map file is *not* required for users to run jQuery, it just improves performance. In 1.11.0/2.1.0 the `/** sourceMappingURL` comment is **not included** in the compressed

To locally download these files, right-click the link and select "Save as..." from the menu.

#### jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version of jQuery.

Download the compressed, production-ready jQuery 3.6.0.

[Download the uncompressed, production-ready jQuery 3.6.0](#)

[Download the map file for jQuery 3.6.0](#)

You can also use the slim build of jQuery 3.6.0.

[Download the compressed, production-ready jQuery 3.6.0 slim build](#)

[Download the uncompressed, production-ready jQuery 3.6.0 slim build](#)

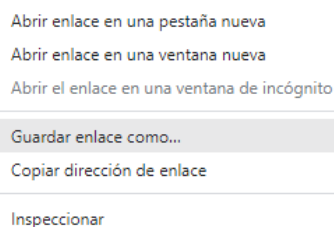


Imagen 2. Descarga jQuery.

Fuente: [jQuery.com](https://jquery.com/).

- **Paso 2:** añade el archivo dentro del directorio de tu proyecto:

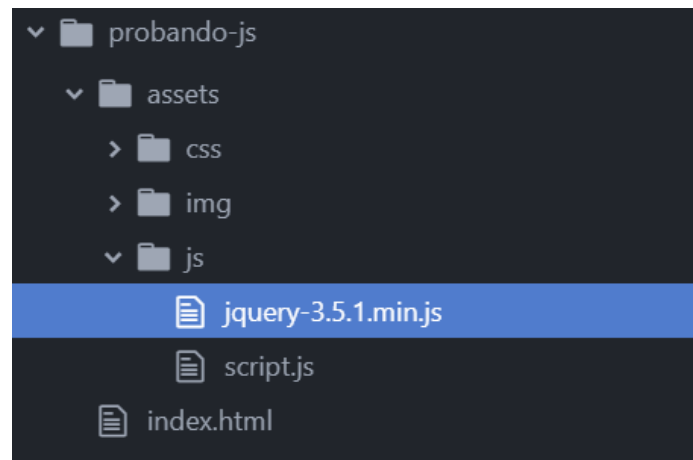


Imagen 3. Descarga jQuery.

Fuente: [jQuery.com](https://jquery.com).

- **Paso 3:** llámalo desde la etiqueta `<script>` con el atributo `src`. Tal como llamaste a un archivo `.js`, ya que toda la biblioteca de jQuery está escrita en JavaScript:

```
<!DOCTYPE html>
<html lang="es" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Probando JS</title>
  </head>
  <body>

    <script src="assets/js/jquery-3.6.0.min.js"></script>
  </body>
</html>
```

## 2. CDN

La otra forma, y la que utilizaremos más a menudo, es usar un CDN desde [Code jQuery](#).

- **Paso 1:** visita la página de [Code jQuery](#) y escoge la versión a utilizar:



Imagen 4. Code jQuery.

Fuente: [jQuery CDN](#).

- **Paso 2:** debes agregar la referencia en la etiqueta `<script>` en tu HTML:

```
<!DOCTYPE html>
<html lang="es" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Probando JS</title>
  </head>
  <body>

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"><
  </script>
  </body>
</html>
```

## Sintaxis jQuery

La sintaxis de jQuery está hecha para seleccionar elementos HTML y realizar alguna acción en el o los elementos seleccionados.

Su sintaxis básica es la siguiente:

```
$(selector).accion()
```



Es una buena práctica esperar a que el documento esté completamente cargado y listo antes de trabajar con él, es por eso que existe el Document Ready Event. Dentro de esa función debemos escribir todas las acciones que necesitemos ejecutar.

A través de las nuevas versiones, esta se ha simplificado hasta quedar así:

```
$(function(){  
  
    // métodos de jQuery...  
  
});
```

Más adelante, profundizaremos más en la sintaxis de jQuery a través de las distintas acciones que podemos hacer.

## Selectores

Los selectores jQuery nos permiten seleccionar y manipular elementos HTML. Se utilizan para "buscar" (o seleccionar) elementos HTML según su nombre, id, clases, tipos, atributos, valores de atributos y mucho más. Se basa en los selectores de CSS existentes y, además, tiene algunos selectores personalizados.

Todos los selectores en jQuery comienzan con el signo \$ y siguen con los paréntesis: \$( ).

Tipos de selectores		
<ul style="list-style-type: none"> <li>Selector universal.</li> </ul>	<ul style="list-style-type: none"> <li>Selector de elementos actual.</li> </ul>	<ul style="list-style-type: none"> <li>Selector por etiqueta.</li> </ul>
<ul style="list-style-type: none"> <li>Selector por id.</li> </ul>	<ul style="list-style-type: none"> <li>Selector por clase.</li> </ul>	<ul style="list-style-type: none"> <li>Selector por atributo.</li> </ul>
<ul style="list-style-type: none"> <li>Selector por valor de atributo.</li> </ul>	<ul style="list-style-type: none"> <li>Selector por número de elementos.</li> </ul>	<ul style="list-style-type: none"> <li>Selector por elemento par.</li> </ul>
<ul style="list-style-type: none"> <li>Selector por elemento impar.</li> </ul>	<ul style="list-style-type: none"> <li>Entre muchos otros.</li> </ul>	

Tabla 1. Tipos de selectores.  
Fuente: Desafío Latam.



Puedes ver el [selector tester de W3Schools](#), donde podrás conocer mejor a los selectores.

Click a selector to see which element(s) gets selected in the result:

```

$( "#LastName" )
$( ".intro" )
$( ".intro, #LastName" )
$( "h1" )
$( "h1, p" )
$( "p:first" )
$( "p:last" )
$( "tr:even" )
$( "tr:odd" )
$( "p:first-child" )
$( "p:first-of-type" )
$( "p:last-child" )
$( "p:last-of-type" )
$( ":nth-child(1)" )
$( ":nth-last-child(1)" )
$( ":nth-of-type(2)" )
$( ":nth-last-of-type(2)" )
$( ":only-child" )
$( ":only-of-type" )
$( "div > p" )
$( "div p" )
$( "ul + p" )
$( "ul :eq(0)" )
$( "ul :gt(0)" )
$( "ul :lt(2)" )
$( ":header" )
$( ":header:not(h1)" )
$( ":animated" )
$( ":focus" )
$( ":contains(Duck)" )
$( "div.has(a)" )
$( ":empty" )
$( ":parent" )
$( ":hidden" )
$( "table:visible" )
$( ":root" )
$( ":lang(it)" )
$( "[id]" )
$( "[id=my-Address]" )
$( "[id!=my-Address]" )
$( "[id=ess]" )
$( "[id=my]" )
$( ":not(a)" )

```

Result:

**Welcome to My Homepage**

```

<h1>
<div class="intro">
<p> My name is Donald <span id="LastName"> Duck </span> </p>
<p id="my-Address"> I live in Duckburg </p>
<p> I have many friends: </p>
</div>
<ul id="ListFriends">
  <li> Goofy </li>
  <li> Mickey </li>
  <li> Daisy </li>
  <li> Pluto </li>
</ul>
<p> I really like Daisy! </p>
<p lang="it" title="Hello beautiful"> Ciao bella </p>
<h3> We are all animals! </h3>
<p> <b> My latest discoveries has led me to believe that we are all animals: </b> </p>
<table>
  <thead>
    <tr>
      <th> Name
      <th> Type of Animal
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> Mickey
      <td> Mouse
    </tr>
    <tr>
      <td> Goofy
      <td> Dog
    </tr>
    <tr>
      <td> Daisy
      <td> Duck
    </tr>
    <tr>
      <td> Pluto
      <td> Dog
    </tr>
  </tbody>
</table>

```

Subscribe to my newsletter:

Name:

Imagen 5. Sección de Selectores.  
Fuente: [Try jQuery Selector](#).





Además, puedes revisar esta [referencia](#) de W3Schools donde enlistan los distintos selectores.



Consultando las referencias será más fácil trabajar con los selectores. Procura seleccionar bien el elemento al que le quieres dar la acción o atribuirle un método.

## Eventos y Funciones

### ¿Qué son los eventos?

Las diferentes acciones que los visitantes pueden hacer dentro de una página web se denominan eventos y representan el momento preciso en que algo sucede.

Ejemplos:

- Cuando el usuario clickea un elemento;
- Cuando el usuario presiona una tecla;
- Cuando el usuario hace scroll;
- Cuando el usuario cambia el tamaño de la ventana de visualización del navegador.

Aquí te presentamos algunos ejemplos de los eventos más usados:

Mouse	Teclado	Formularios	Documento/Ventana
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
hover			

Tabla 2. Eventos.  
Fuente: Desafío Latam.

## Sintaxis de los eventos

La sintaxis es la siguiente:

```
$(selector).evento(function(){  
    // La acción ocurre aquí  
});
```

Como vemos, es necesario pasarle una función al evento, que se ejecutará cuando el evento suceda. Ejemplo:

```
$("p").dblclick(function(){  
    $(this).hide();  
});
```

Por ejemplo, en el siguiente código, jQuery hará que los elementos `<p>` se oculten, si el usuario les da doble click:

```
<body>  
<p>If you double-click on me, I will disappear.</p>  
<p>Click me away!</p>  
<p>Click me too!</p>  
  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"><  
</script>  
<script>  
$(document).ready(function(){  
    $("p").dblclick(function(){  
        $(this).hide();  
    });  
});  
</script>  
</body>
```

Puedes ejecutar este código tú mismo, desde el [editor de W3Schools](https://www.w3schools.com/js/tryit.asp?filename=try_js_events_click).

## Métodos

Los métodos son la acción a la que se ve enfrentada el selector y son diferentes a los eventos:

```
$(selector).accion()
```

Un evento es el resultado de una acción realizada por el usuario, como hacer click, desplazar, arrastrar, redimensionar, etc. En cambio, un método es la acción a la que se ve enfrentado, lo que pueden ser concatenados y afectar al selector. Muchas veces estos métodos son activados cuando un evento ocurre (como lo que hicimos en el ejercicio anterior).

Existen muchos métodos de jQuery, y podemos dividirlos en los siguientes tipos:

- **Métodos de efectos:** Proporcionan varias técnicas para agregar animación a una página web, incluyendo efectos simples y personalizados.
- **Métodos de manipulación de HTML y CSS:** Manipulan los elementos, cambiando sus atributos, estilos y/o propiedades.

Veamos algunos de ellos:

## Efectos

- `animate();`: realiza una animación personalizada de un conjunto de propiedades CSS.
- `delay();`: configura un temporizador para retrasar la ejecución de los elementos siguientes en la cola.
- `fadeIn();`: muestra los elementos coincidentes, atenuándolos a opacos.
- `fadeOut();`: oculta los elementos coincidentes, atenuándolos a transparentes.
- `fadeTo();`: ajusta la opacidad de los elementos seleccionados.
- `fadeToggle();`: muestra u oculta los elementos seleccionados, animando su opacidad

- `finish()`; detiene la animación que se está ejecutando, elimina todas las animaciones en cola y completa todas las animaciones para los elementos seleccionados.
- `hide()`; oculta los elementos coincidentes.
- `show()`; muestra los elementos seleccionados.
- `stop()`; detiene la animación que se está ejecutando actualmente en los elementos coincidentes.
- `toggle()`; muestra u oculta los elementos coincidentes.

Para ver la lista completa de efectos, puedes consultar la [documentación de jQuery](#):

## Manipulación de HTML y CSS

- `addClass()`; agrega la clase especificada a cada elemento seleccionado.
- `after()`; inserta contenido, especificado por parámetro, después de cada elemento seleccionado.
- `append()`; inserta contenido, especificado por parámetro, al final de cada elemento seleccionado.
- `appendTo()`; inserta todos los elementos seleccionados hasta el final del objetivo definido por parámetro.
- `attr()`; obtiene el valor de un atributo para el primer elemento seleccionado o establece uno o más atributos para cada elemento coincidente.
- `before()`; inserta contenido, especificado por parámetro, antes de cada elemento seleccionado.
- `clone()`; crea una copia profunda del conjunto de elementos coincidentes.
- `css()`; obtiene el valor de una propiedad de estilo para el primer elemento seleccionado o establece una o más propiedades CSS para cada elemento seleccionado.
- `detach()`; elimina el conjunto de elementos coincidentes del DOM.

- **empty();** elimina todos los nodos secundarios del conjunto de elementos seleccionados del DOM.
- **hasClass();** determina si alguno de los elementos seleccionados posee la clase dada.
- **height();** obtiene la altura calculada actual para el primer elemento del conjunto de elementos seleccionados o establece la altura de cada elemento coincidente.
- **html();** obtiene el contenido HTML del primer elemento del conjunto de elementos seleccionados o establece el contenido HTML de cada elemento coincidente.
- **remove();** elimina el conjunto de elementos seleccionados del DOM.
- **removeAttr();** elimina un atributo de cada elemento del conjunto de elementos seleccionados.
- **removeClass();** elimina una sola clase, varias clases o todas las clases de cada elemento en el conjunto de elementos seleccionados.
- **text();** obtiene el contenido de texto combinado de cada elemento en el conjunto de elementos seleccionados, incluidos sus descendientes, o establece el contenido de texto de los elementos coincidentes.
- **toggleClass();** agrega o elimina una o más clases de cada elemento en el conjunto de elementos seleccionados.
- **val();** obtiene el valor actual del primer elemento en el conjunto de elementos seleccionados o establece el valor de cada elemento coincidente.
- **width();** obtiene el ancho calculado actual para el primer elemento en el conjunto de elementos coincidentes o establezca el ancho de cada elemento coincidente.

Para ver la lista completa de métodos, puedes consultar la [documentación de jQuery](#).



**Antes de continuar:**

¿Existe algún concepto que no hayas comprendido?

Vuelve a revisar los conceptos que más te hayan costado antes de seguir adelante.

## Ejercicio guiado: Primeros pasos con jQuery

### Contexto

Al igual que hicimos con JavaScript puro, es importante que practiquemos de forma aislada la sintaxis y algunos métodos de jQuery, para entender su alcance y experimentar la utilidad que nos ofrecen. Para esto, vamos a probar algunos métodos y eventos de jQuery.

Sigue el paso a paso que te presentamos a continuación:

- **Paso 1:** crea una carpeta llamada probando-jquery con la estructura que ya conocemos:

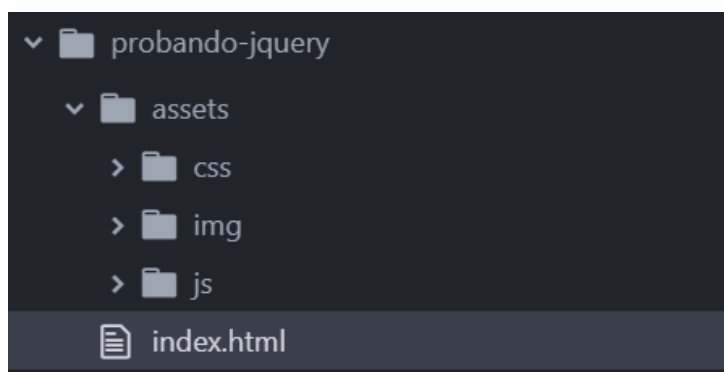


Imagen 6. Probando jQuery.  
Fuente: Desafío Latam.

- **Paso 2:** integra jQuery por CDN, para eso, en la estructura del HTML agregaremos un tag `<script>` para agregar al enlace que podemos obtener [aquí](https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js):

```
<!DOCTYPE html>
<html lang="es" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Probando jQuery</title>
  </head>
  <body>

    <script
      src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
    </script>

  </body>
```

```
</html>
```

Esto permitirá integrar elementos de jQuery en el archivo. Prueba algunos métodos.

- **Paso 3:** copia el siguiente código en el documento `index.html` y ejecútalo en el navegador:

```
<body>
  <p>Si me clickeas dos veces, desapareceré.</p>
  <p>¡Yo también!</p>
  <p>¡Yo también!</p>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"><
  /script>
  <script>
    $(document).ready(function(){
      $("p").dblclick(function(){
        $(this).hide();
      });
    });
  </script>
</body>
```

- **Paso 4:** observa el comportamiento. Si activas el evento `dblclick()` en cualquiera de los textos, estos activarán el efecto `hide()` y desaparecerán de manera independiente:



Imagen 7. Primera ejecución con jQuery.  
Fuente: Desafío Latam.

- **Paso 5:** si buscas la [documentación](#) de `hide()`, observarás que, además, puede recibir parámetros para controlar su comportamiento, como la duración, transición, etc. Prueba controlando la duración de la animación, con el parámetro `'slow'`

El script quedará de la siguiente manera:

```
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide("slow");
  });
});
</script>
```

Guarda y observa el resultado.



Otra opción para controlar la duración de la animación es `"fast"`. Comprueba la diferencia.

- **Paso 6:** ahora prueba algo diferente, modifica el CSS a través de jQuery. Cuando se haga un click sobre el texto, aumentará su tamaño y cambiará a color rojo, de la siguiente manera:

```
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide('slow');
  });

  $("p").click(function(){
    $(this).css({
      "color": "red",
      "font-size": "2em"
    });
  });
});
</script>
```



Guarda y observa el resultado:



Imagen 8. Modificando el CSS.  
Fuente: Desafío Latam.

- **Paso 7:** además del formato, puedes cambiar el contenido del HTML. Por ejemplo, que al pasar el mouse sobre cualquier párrafo, este se transforme en un link y cambie su contenido:

```
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide('slow');
  });

  $("p").click(function(){
    $(this).css({
      "color": "red",
      "font-size": "2em"
    });
  });

  $("p").hover(function(){
    $(this).html("<a href='#'>Me convertí</a>");
  });
});
</script>
```

Guarda y valida el resultado:

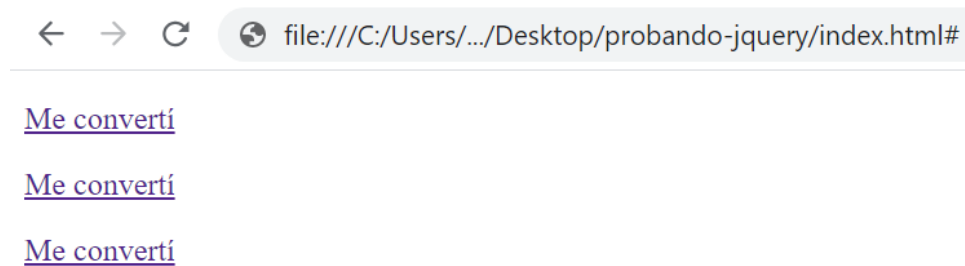


Imagen 9. Modificando el contenido.  
Fuente: Desafío Latam.

## Resumen

- jQuery es una biblioteca de JavaScript que permite escribir menos código, además, hace más sencilla la captura y manipulación de los diferentes elementos de un sitio, encapsulando el código JavaScript en métodos más cortos y simples.
- Se puede integrar jQuery a un proyecto descargando el archivo o usando un CDN desde [Code jQuery](#).
- La sintaxis de jQuery es la siguiente: `$(selector).accion()`
- Los selectores jQuery permiten seleccionar y manipular elementos HTML.
- Todos los selectores en jQuery comienzan con el signo `$` y siguen con los paréntesis: `$()`.
- Los eventos son las diferentes acciones que los visitantes pueden hacer dentro de una página web se denominan y representan el momento preciso en que algo sucede.
- Los métodos son la acción a la que se ve enfrentada el selector.