



# APIs

Manipulación de elementos del DOM con jQuery y eventos



***Crear plugins propios con  
jQuery para ampliar las  
funcionalidades de una  
aplicación web.***

- Unidad 1:  
Introducción al lenguaje JavaScript
- Unidad 2:  
Funciones y Ciclos
- Unidad 3:  
Arrays y Objetos
- Unidad 4:  
APIs



Te encuentras aquí



## ¿Qué aprenderás en esta sesión?

- *Desarrolla algoritmos transformando código de JavaScript a jQuery para utilizar correctamente la sintaxis de la librería.*
- *Crea un script que manipule elementos del DOM con selectores utilizando la librería jQuery para resolver un problema planteado.*
- *Crea un script que permita la selección y manipulación de elementos del DOM utilizando la librería jQuery para resolver un problema planteado.*

# ***/\* Selección de elementos del DOM con jQuery \*/***

# Filtros con jQuery

- Los filtros son selectores especiales de jQuery que permiten acotar el espectro de una selección, limitado a la condición que se especifica en dicho filtro. Los filtros se verán marcados con el símbolo dos puntos (:).
- En esta sesión veremos 3 tipos de filtros:



# Filtros con jQuery

## Filtros de posición

`:gt(1)`



Retorna los elementos que sean mayores al índice 1 presentes en la lista. Excluye a los elementos que están en la posición 0 y 1.

`:first`



Retorna el primer elemento de los existentes en los ítems de la lista.

`eq(n)`



(Equal to)  
Devuelve el elemento que está en el índice n.

`lt(n)`



(Less than)  
Devuelve los elementos que están en el índice menor que el valor n.

# Demostración - “Filtros con jQuery”



# Ejercicio guiado

## Filtros con jQuery

Implementar los filtros de posición en una tabla con tres filas y tres columnas, se debe seleccionar y modificar el color de fondo de la primera y última fila, además del color del texto de la primera fila y el elemento de la columna tres con fila dos. El HTML a continuación será el punto de partida para el ejercicio.

```
<table>
  <tr>
    <td>Row 1</td>
    <td>Row 2</td>
    <td>Row 3</td>
  </tr>
  <tr>
    <td>Row 4</td>
    <td>Row 5</td>
    <td>Row 6</td>
  </tr>
  <tr>
    <td>Row 7</td>
    <td>Row 8</td>
    <td>Row 9</td>
  </tr>
</table>
```



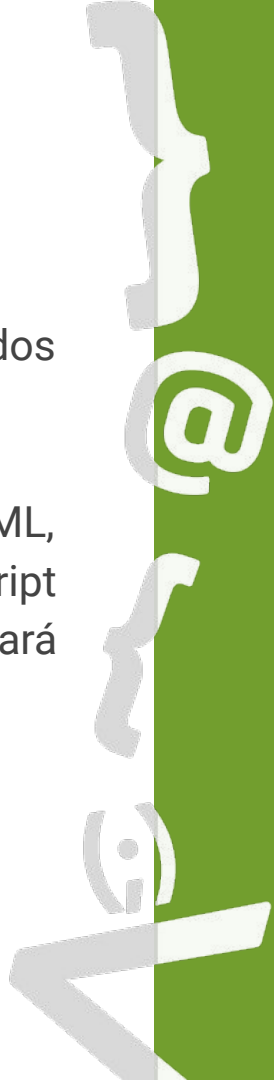


# Ejercicio guiado

## *Filtros con jQuery*

**Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.

**Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y agregar el extracto de HTML indicado al inicio del ejercicio. Además, se agregará un estilo con CSS para darle bordes a la tabla y así visualizar mejor el ejemplo:



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <style>
    table, tr, td {
      border: 1px solid black;
    }
  </style>
  <title>Document</title>
</head>
<body>
  <table>
    <tr>
      <td>Row 1</td>
      <td>Row 2</td>
      <td>Row 3</td>
    </tr>
    <tr>
      <td>Row 4</td>
      <td>Row 5</td>
      <td>Row 6</td>
    </tr>
    <tr>
      <td>Row 7</td>
      <td>Row 8</td>
      <td>Row 9</td>
    </tr>
  </table>
  <script src="script.js"></script>
</body>
</html>
```

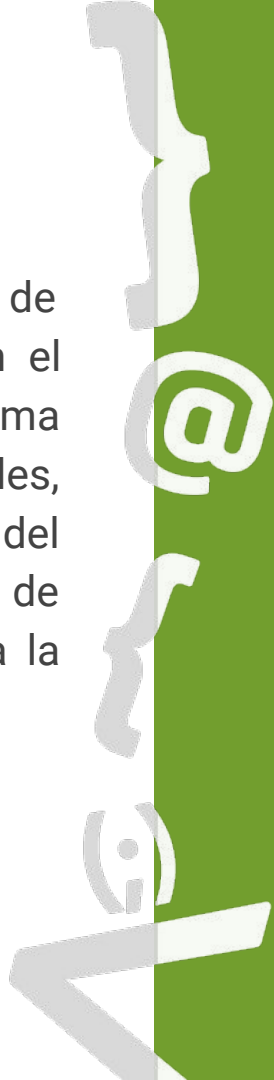


# Ejercicio guiado

## Filtros con jQuery

- **Paso 3:** En el archivo script.js se trabajará con los selectores del tipo filtro de jQuery para poder realizar las modificaciones pertinentes solicitadas en el enunciado. Una de ellas, es el cambio de color de fondo de la primera y última fila. Esto se puede lograr implementado los filtros lt y gt, los cuales, seleccionan todos los elementos en un índice mayor o menor (dependiendo del filtro) que el índice dentro del conjunto coincidente. Para cambiar el color de fondo se utilizará el método de jQuery denominado CSS, donde se pasa la propiedad más el valor.

```
$(document).ready(function() {  
    $('td:gt(5)').css('backgroundColor', 'blue');  
    $('td:lt(3)').css('backgroundColor', 'yellow');  
});
```

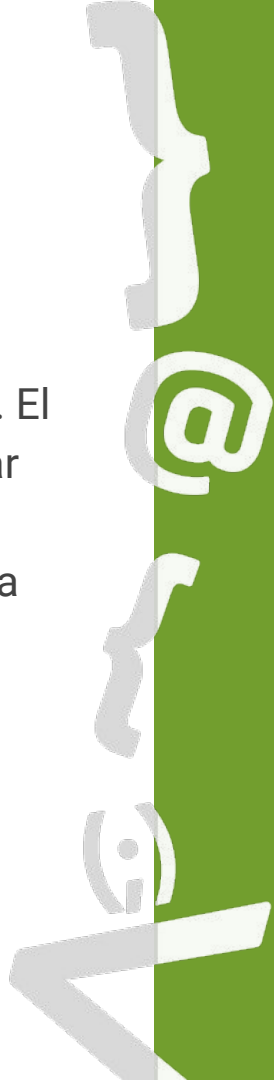


# Ejercicio guiado

## Filtros con jQuery

- **Paso 4:** Para poder terminar el ejercicio, aún hace falta modificar el color del texto de la primera fila y la fila dos, pero en la columna tres específicamente. El primer caso se puede lograr mediante el filtro “:first”, que permite seleccionar el primer elemento del DOM que coincida con el selector indicado, mientras que para seleccionar y modificar el elemento de la segunda fila con la tercera columna, se puede implementar el filtro “:eq”.

```
$(document).ready(function() {  
    $('td:gt(5)').css('backgroundColor', 'blue');  
    $('td:lt(3)').css('backgroundColor', 'yellow');  
    $('tr:first').css('color', 'red');  
    $('td:eq(5)').css('color', 'red');  
});
```



# Filtros con jQuery

## *Filtros hijo*

- Para implementar los filtros del tipo hijo con jQuery, se debe utilizar la pseudoclase **“last-child”**.
- Por ejemplo, si se tiene una estructura de listas no ordenadas y se desea seleccionar de esa lista solamente el último elemento y mostrar el texto que ese elemento posee, se debe realizar

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ul>
```

```
$(document).ready(function() {
  $('ul li:last-child').html();
});
```

# Demostración - “Filtros hijo”



# Ejercicio guiado

## Filtros hijo

Se solicita seleccionar y modificar con el método CSS de jQuery los nombres de “Maria y Paola”. La modificación debe ser del color de fondo y tamaño del texto. Siendo el extracto del HTML el siguiente:

```
<div>
  <span>Juan,</span>
  <span>Carlos,</span>
  <span>Manuel,</span>
  <span>María</span>
</div>
<div>
  <span>Rosa,</span>
  <span>Daniela,</span>
  <span>Pedro,</span>
  <span>Paola</span>
</div>
```



# Ejercicio guiado

## Filtros hijo

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.
- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar el trozo de código mostrado en el enunciado del ejercicio:





```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <title>Document</title>
</head>
<body>
  <div>
    <span>Juan,</span>
    <span>Carlos,</span>
    <span>Manuel,</span>
    <span>María</span>
  </div>
  <div>
    <span>Rosa,</span>
    <span>Daniela,</span>
    <span>Pedro,</span>
    <span>Paola</span>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

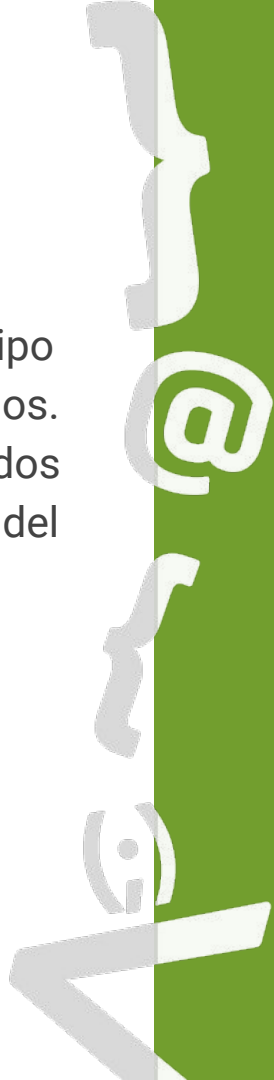


# Ejercicio guiado

## Filtros hijo

- **Paso 3:** Ahora en el archivo script.js, se trabajará con los selectores de tipo filtro hijo de jQuery para poder realizar la selección de los elementos indicados. En este caso, consiste en seleccionar y modificar con el método CSS los dos últimos elementos de cada etiqueta <div>. Esto se puede lograr con el filtro del tipo “:last-child”, aplicado a la etiqueta <div>.

```
$(document).ready(function() {  
    $('div  
span:last-child').css('backgroundColor', 'blue');  
});
```



# Filtros con jQuery

## *Filtros form*

- También es posible añadir filtros a elementos del DOM de tipo form. Estos filtros son útiles para capturar información de un usuario mientras escribe algún valor en un formulario, cambia el estado de cierto elemento o simplemente desea enviar dichos datos a algún servidor para ser procesados.

```
$(document).ready(function() {  
    $('input[type="checkbox"]:checked');  
});
```

# Demostración - “Filtros form”



# Ejercicio guiado

## Filtros form

Se solicita acceder y mostrar el elemento que ya viene seleccionado por defecto en la etiqueta `<option>`:

```
<select name="países">
  <option value="Argentina">Argentina</option>
  <option value="Chile" selected>Chile</option>
  <option value="Brasil">Brasil</option>
</select>
```

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crear dos archivos, un `index.html` y un `script.js`.



# Ejercicio guiado

## Filtros form

**Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio:

{desafío}  
latam\_

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <script
src="https://code.jquery.com/jquery-3.7.1.min
.js"></script>
  <title>Document</title>
</head>
<body>
  <select name="países">
    <option
value="Argentina">Argentina</option>
    <option
value="Chile">Chile</option>
    <option
value="Brasil">Brasil</option>
  </select>
  <script src="script.js"></script>
</body>
</html>
```

# Ejercicio guiado

## Filtros form

- **Paso 3:** Utilizando un filtro form, debemos acceder al elemento y filtrar por el selector principal ("select"), luego accedemos al hijo mediante el selector ">", siendo el elemento "option" el hijo directo, aquí si se aplica el filtro "selected", posteriormente con el método "val" capturamos el valor (value) que tiene el "option".

```
$(document).ready(function() {  
    $('select > option:selected').val();  
});
```



# Ejercicio guiado

## Filtros form

- **Paso 4:** El método “val” nos ayuda a obtener el valor del elemento, por ende, guardemos el resultado en una variable y usamos console.log para mostrar el valor.

```
$(document).ready(function() {  
    let valorDeLaOpcion = $('select > option:selected').val();  
    console.log(valorDeLaOpcion)  
});
```

- El resultado por consola es:

Chile





## Otros filtros interesantes

La gama de filtros es enorme y la cantidad de acciones que se pueden realizar con ellos es compleja. Existen filtros para :

- obtener elementos animados (:animated)
- elementos ocultos (:hidden)
- un elemento raíz (:root)
- elementos visibles (:visible).

La gama de filtros siempre la puedes consultar al detalle en la documentación oficial de [jQuery](#).

**`/* Eventos con jQuery */`**

# ¿Qué es un evento?

- Es una acción o función determinada que se ejecuta a partir de una interacción con nuestro sitio web. Por ejemplo, el hacer click sobre un botón o pasar el mouse sobre algún elemento html.

```
$(document).ready(function() {  
    $('button').click(function () {  
        alert("hola mundo");  
    });  
});
```

- En general, los eventos en jQuery los podemos identificar o necesitar cuando queremos "hacer algo" a partir de una acción.

# Demostración - “Eventos con jQuery”



# Ejercicio guiado

## Eventos con jQuery

Imagina que en una página web solicitan que se visualice en un párrafo la selección que ha hecho el usuario en un formulario, mediante la etiqueta `<select>`, es decir, que si se elige Chile como opción dentro de la lista, este texto lo veamos reflejado en alguna parte del sitio.

Mediante jQuery y la captura de eventos, vamos a recoger esta información utilizando el método `change`, en base al siguiente código:

```
<p class="resultado"></p>
<select name="pais" id="pais">
  <option
value="Argentina">Argentina</option>
  <option value="Perú">Perú</option>
  <option value="Chile">Chile</option>
  <option
value="Venezuela">Venezuela</option>
</select>
```

# Ejercicio guiado

## Eventos con jQuery

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.
- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio:

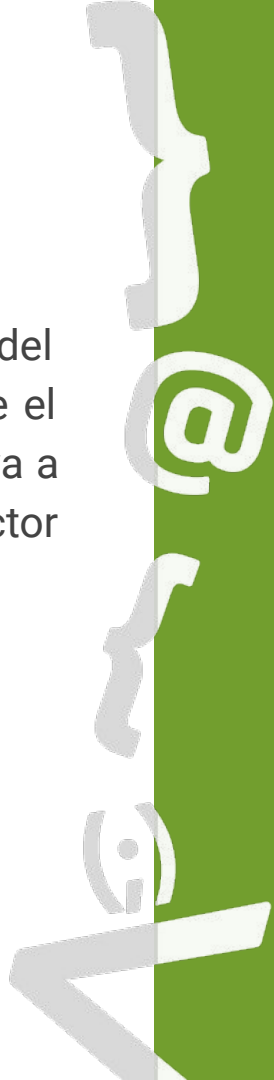
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <script
src="https://code.jquery.com/jquery-3.7.1.min.js"></s
cript>
  <title>Document</title>
</head>
<body>
  <p class="resultado"></p>
  <select name="pais" id="pais">
    <option value="Argentina">Argentina</option>
    <option value="Perú">Perú</option>
    <option value="Chile">Chile</option>
    <option value="Venezuela">Venezuela</option>
  </select>
  <script src="script.js"></script>
</body>
</html>
```

# Ejercicio guiado

## Eventos con jQuery

- **Paso 3:** Lo primero que haremos será acceder al elemento principal del formulario, es decir, a la etiqueta “select” que posee un id="pais", mediante el selector de jQuery: \$('#pais'). Aplicaremos el evento change, el cual, conlleva a una función que se ejecutará cada vez que ocurra ese evento en el selector indicado al método.

```
$(document).ready(function() {  
    $('#pais').change(function() {  
    })  
});
```

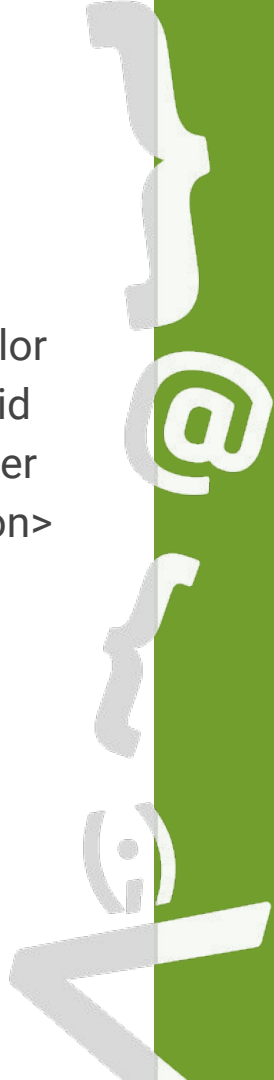


# Ejercicio guiado

## Eventos con jQuery

- **Paso 4:** Ahora necesitamos acceder a la opción seleccionada, obtener su valor y mostrarlo en el elemento p. Eso se logra indicando que al elemento con el id igual a “pais”, es decir, el “select”, se le aplique el método “children” para poder capturar el hijo que fue seleccionado, siendo el hijo directo la etiqueta <option> y en ella, activamos el filtro “:selected” más el método “val” para obtener el valor.

```
$(document).ready(function() {  
    $('#pais').change(function() {  
        let paisSeleccionado = $(this).children("option:selected").val();  
        $('#resultado').text(paisSeleccionado);  
    })  
});
```





# Utilizando eventos con jQuery

- Uno de los métodos más implementado para manejar cualquier tipo de evento, es el **método “.on()”**, el cual, permite indicar el tipo de evento y agregar una función para el manejo de la acción.
- En el siguiente código se utiliza el método “.on()” para capturar el evento “click” y mostrar por consola un mensaje:

```
$(document).ready(function() {  
    $( "button" ).on( "click", function() {  
        console.log( "Ocurrió un clic sobre la etiqueta p" );  
    });  
});
```

# Demostración - “Seleccionar y cambiar el color de fondo”



# Ejercicio guiado

## *Seleccionar y cambiar el color de fondo*

Se solicita seleccionar y cambiar el color de fondo a los elementos impares dentro de la lista desordenada cuando el usuario haga un click en el botón. Para ello se tiene el siguiente código en html:

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
  <li>item 4</li>
  <li>item 5</li>
  <li>item 6</li>
</ul>
<button id="btn">Seleccionar</button>
```



# Ejercicio guiado

## Seleccionar y cambiar el color de fondo

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.
- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio:

{desafío}  
latam\_

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <script
src="https://code.jquery.com/jquery-3.7.1.min
.js"></script>
  <title>Document</title>
</head>
<body>
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
    <li>item 4</li>
    <li>item 5</li>
    <li>item 6</li>
  </ul>
  <button id="btn">Seleccionar</button>
  <script src="script.js"></script>
</body>
</html>
```

# Ejercicio guiado

## Seleccionar y cambiar el color de fondo

- **Paso 3:** Para dar solución a lo solicitado, necesitamos usar algún evento de jQuery que permita capturar el evento “click” cuando active el botón. En este caso, uno de los posibles métodos que se pueden emplear, es el “.on()”, quién podrá ser aplicado directamente al botón, quedando:

```
$(document).ready(function() {  
    $("#btn").on("click",function() {  
        console.log( "Ocurrió un clic sobre el botón" );  
    });  
});
```

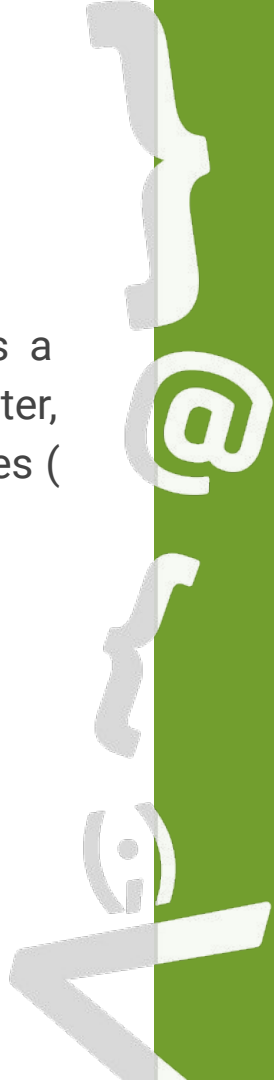


## Ejercicio guiado

### *Seleccionar y cambiar el color de fondo*

- **Paso 4:** Ahora como ya está activo y trabajando el evento, procedemos a seleccionar los elementos “li” en posiciones impares mediante el método filter, y con el pseudo elemento :even podemos seleccionar las posiciones impares ( y con :odd las pares).

```
$( "li" ).filter( ":even" );
```



# Ejercicio guiado

## Seleccionar y cambiar el color de fondo

- **Paso 5:** Como ya seleccionamos los elementos, realicemos alguna acción sobre ellos, por ejemplo, cambiar el color de fondo:

```
$(document).ready(function() {  
    $("#btn").on("click",function() {  
        $( "li" ).filter( ":even" ).css( "background-color", "red" );  
    });  
});
```

- **Paso 6:** Si ejecutamos el código anterior, y hacemos un click en el botón, la selección ocurrirá sobre los ítem con número impar.



# Demostración - “Contar los clicks sobre un elemento”

{desafío}  
latam\_





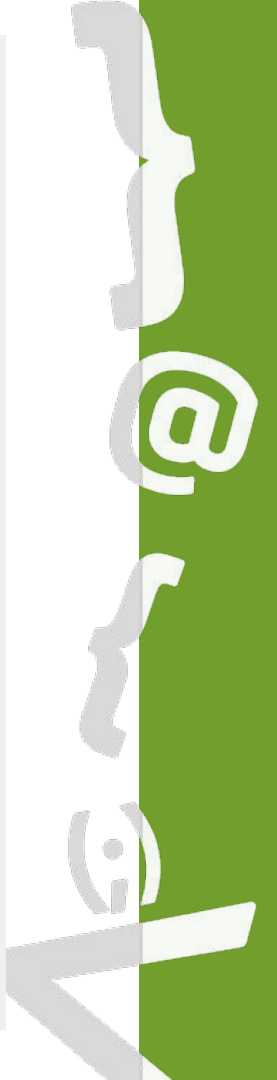
# Ejercicio guiado

## Seleccionar y cambiar el color de fondo

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.
- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio:

{desafío}  
latam\_

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <script
src="https://code.jquery.com/jquery-3.7.1
.min.js"></script>
    <title>Ejemplo jQuery</title>
  </head>
  <body>
    <h3>Clic en la imagen</h3>
    
    <script src="script.js"></script>
  </body>
</html>
```

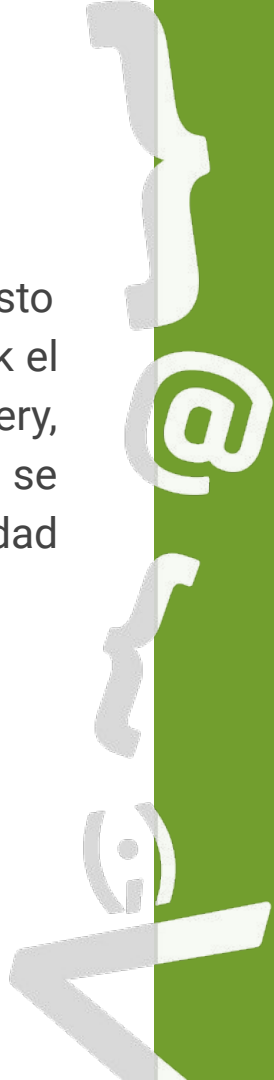


# Ejercicio guiado

## Seleccionar y cambiar el color de fondo

- **Paso 3:** En el archivo script.js, crear una variable para llevar un contador, esto nos servirá para tener presente la cantidad de veces que está haciendo click el usuario sobre la imagen. Luego, utilizando el selector de id mediante jQuery, vamos a capturar el evento “click” sobre la imagen y dentro de la función se realizará el aumento del contador en uno y el mensaje que indicará la cantidad de clic que se llevan sobre la imagen.

```
$(document).ready(function() {  
    let contador = 0;  
    $('#ejemplo').on('click', function() {  
        contador++;  
        console.log('---> Clic '+contador+' vez!');  
    });  
});
```



- **Paso 4:** Al ejecutar el código anterior en el navegador web y si el usuario hace un click sobre la imagen, el resultado por consola sería:

```
---> Clic 1 vez!
```

- **Paso 5:** Funcionando hasta el momento el código realizado, procedemos a trabajar con una función externa que nos permitirá regresar la hora, minutos y segundos exactamente cuando el usuario haga un click sobre la imagen. Esto se puede lograr implementando el objeto Date() de JavaScript.



```
$(document).ready(function() {  
    function formatDate(date) {  
        return (date.getHours() < 10 ? '0' : '') +  
date.getHours() + ':' + (date.getMinutes() < 10 ? '0' : '') +  
date.getMinutes() + ':' + (date.getSeconds() < 10 ? '0' : '')  
+ date.getSeconds() + '.' + (date.getMilliseconds() < 10 ?  
'00' : (date.getMilliseconds() < 100 ? '0' : '')) +  
date.getMilliseconds();  
    };  
  
    let contador = 0;  
    $('#ejemplo').on('click', function() {  
        contador++;  
        console.log('--->' + formatDate(new Date()) + '  
Clic '+contador+' vez!');  
    });  
});
```

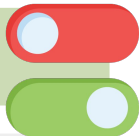
```
--->18:30:19.483 Clic 1 vez!  
--->18:31:01.003 Clic 2 vez!
```



**/\* Animaciones \*/**

# Animaciones

## Toggle



Permite alternar entre 2 acciones

## FadeIn



Cambia gradualmente la opacidad, para los elementos seleccionados, de ocultos a visibles.

## FadeOut



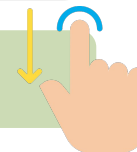
Cambia gradualmente la opacidad, para los elementos seleccionados, de visible a oculto

## slideUp



El método `slideUp()` desliza hacia arriba (oculta) los elementos seleccionados.

## slideDown



El método `slideDown()` desliza hacia abajo (muestra) los elementos seleccionados.

# Demostración - “Toggle”



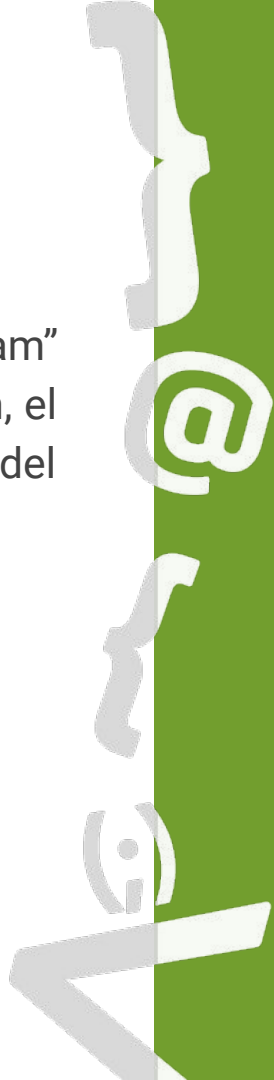
# Ejercicio guiado

## Toggle

Desarrollar una aplicación que permita modificar el texto existente por “Latam” cuando el usuario haga click sobre un botón y, cuando vuelva a realizar la acción, el texto debe regresar a su condición original, en este caso “Desafío”. El extracto del HTML es el siguiente:

```
<p>Desafío</p>  
<p style="display: none">Latam</p>  
<button>Cambio</button>
```

{desafío}  
latam\_





# Ejercicio guiado

## Toggle

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.
- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio, como se muestra a continuación:

{desafío}  
latam\_

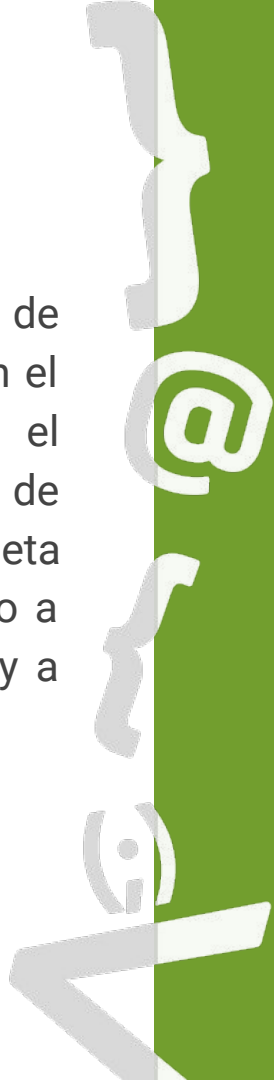
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <script
src="https://code.jquery.com/jquery-3.7.
1.min.js"></script>
    <title>Ejemplo jQuery</title>
  </head>
  <body>
    <p>Desafío</p>
    <p style="display:
none">Latam</p>
    <button>Cambio</button>
    <script src="script.js"></script>
  </body>
</html>
```

# Ejercicio guiado

## Toggle

- **Paso 3:** En el archivo script.js, activar la función principal de jQuery y dentro de ella capturamos el elemento <button> y le asignamos un evento “click” con el método “.on()”. Ahora dentro del método “.on()”, se debe seleccionar el elemento <p> y aplicar el método “.toggle()”. Este método se encarga de modificar el estilo en línea que posean las etiquetas, como la segunda etiqueta <p> contiene un estilo con “display: none”, el toggle se encarga de llevarlo a block, mientras que al otro elemento <p>, le modifica la propiedad display a none.

```
$(document).ready(function() {  
    $('button').on('click',function(){  
        $('p').toggle();  
    });  
});
```



¿Existe algún concepto que no  
hayas comprendido?

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





## Próxima sesión...

- *Desarrollaremos el material sincrónico que corresponde a una **Guía de ejercicios** con la cual practicarás todo lo aprendido en estas sesiones.*

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

