

Desarrollo Web Frontend



Edición 2021

Universidad Autónoma de Entre Ríos
Facultad de Ciencia y Tecnología - Sede Oro Verde



Manipulación del DOM con JQuery

Agregar y eliminar elementos del DOM con JQuery

Manipulando clases CSS con JQuery

Introducción a AJAX con JQuery





Manipulación del DOM con JQuery

Una de las virtudes principales de JQuery es la posibilidad de manipular elementos del DOM. Existe métodos que permiten fácilmente acceder y manipular elementos y atributos.

A continuación veremos tres maneras de manipular elementos del DOM:

- **text()** → define o devuelve el valor del elemento seleccionado.
- **html()** → define o devuelve el valor del elemento seleccionado. Y además Incluye el marcado HTML.
- **val()** → define o devuelve el valor (*value*) de los elementos del formulario.

Ejemplo:

\$(selector).text() - **\$(selector).html()** - **\$(selector).val()**





Manipulación del DOM con JQuery (Continuación...)

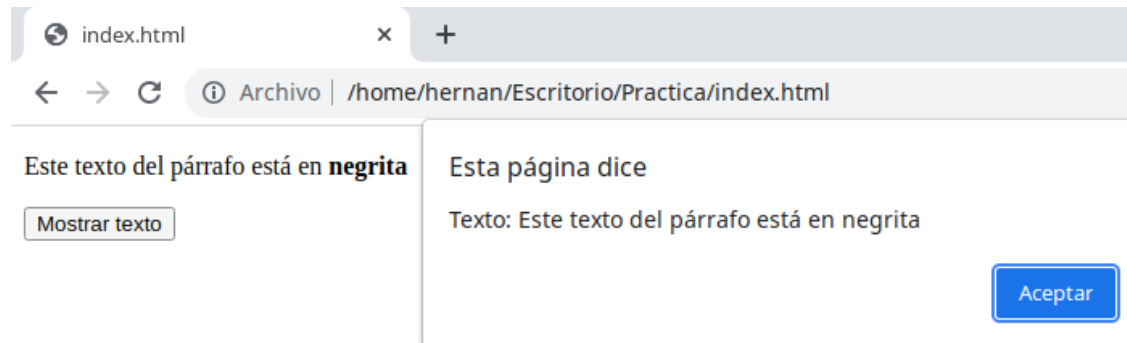
El método text()

A continuación veremos ejemplos de cómo definir y obtener el contenido de un elemento del DOM con el método jQuery **text()**.

Ejemplo: obteniendo el contenido de texto del elemento párrafo con **id="test"**

```
js > JS index.js > ...  
1 $(document).ready(function(){  
2     $("#btn1").click(function(){  
3         alert("Texto: " + $("#test").text());  
4     });  
5 });
```

```
<? index.html > ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="utf-8">  
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
6     <script src="js/index.js?1"></script>  
7 </head>  
8 <body>  
9     <p id="test">Este texto del párrafo está en <b>negrita</b></p>  
10    <button id="btn1">Mostrar texto</button>  
11    <button id="btn2">Mostrar en HTML</button>  
12 </body>  
13 </html>
```





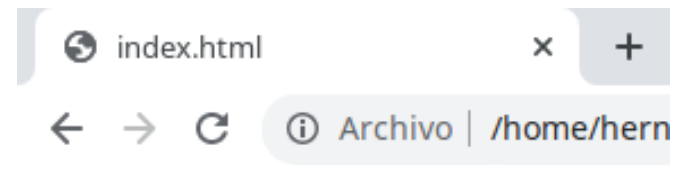
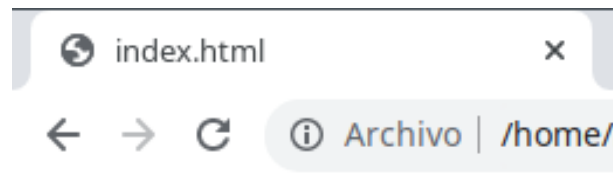
Manipulación del DOM con JQuery (Continuación...)

El método text()

Ejemplo: definimos el contenido de texto del elemento párrafo con **id="test"** “

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("#test").text("El texto se genero al presionar el boton");
4   });
5 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1">
8 </script>
9 </head>
10 <body>
11   <p id="test"></p>
12   <button id="btn1">Mostrar texto</button>
13 </body>
14 </html>
```





Manipulación del DOM con JQuery (Continuación...)

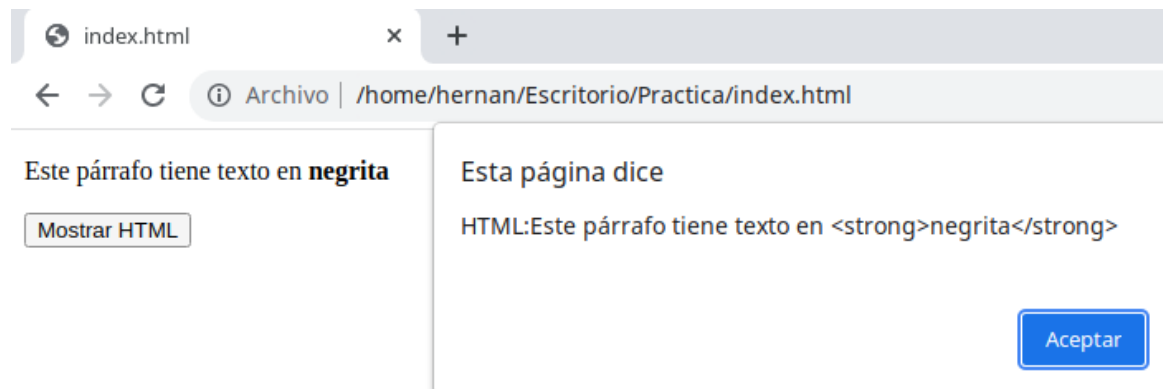
El método html()

Ejemplo: obtenemos el contenido HTML del párrafo con **id="test"**

```
js > JS index.js > ...
```

```
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("#test").html();
4   });
5 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1">
8 </script>
9 </head>
10 <body>
11   <p id="test">Este párrafo tiene texto en <strong>negrita</strong></p>
12   <button id="btn1">Mostrar HTML</button>
13 </body>
14 </html>
```





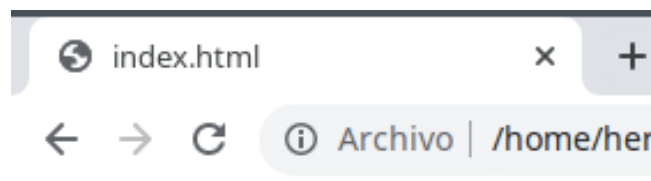
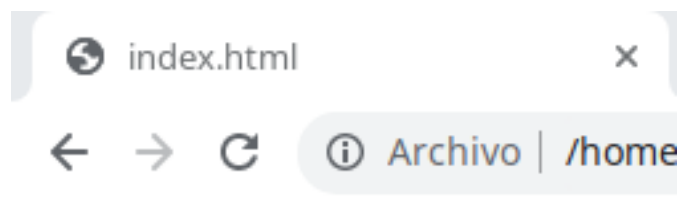
Manipulación del DOM con JQuery (Continuación...)

El método html()

Ejemplo: definiendo el contenido HTML del párrafo con **id="test"**

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("#test").html("Este contenido se genera desde <strong>JQuery</strong>")
4   });
5 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1"></script>
8 </head>
9 <body>
10  <p id="test"></p>
11  <button id="btn1">Mostrar HTML</button>
12 </body>
13 </html>
```



Este contenido se genera desde JQuery

Mostrar HTML





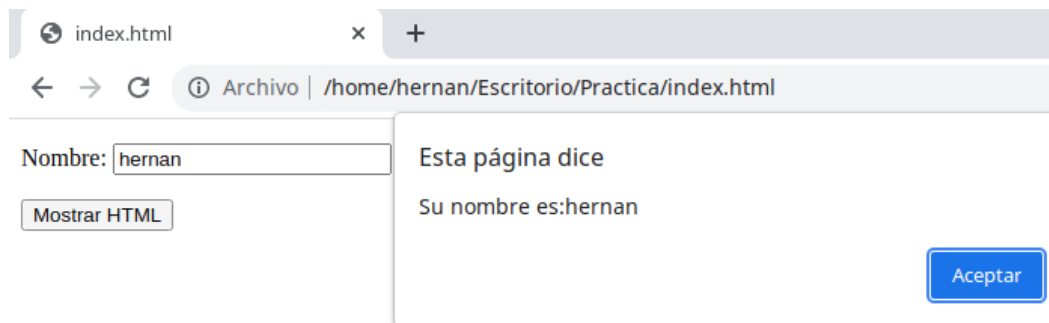
Manipulación del DOM con JQuery (Continuación...)

El método val()

Ejemplo: obteniendo el valor del elemento INPUT con **id="test"**

```
JS index.js  x  JS index copy.js  <> index.html  <> index co
js > JS index.js > ...
1  $(document).ready(function(){
2    $("#btn1").click(function(){
3      alert("Su nombre es:" + $("#test").val());
4    });
5  });
```

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5
6    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/
   jquery.min.js"></script>
7    <script src="js/index.js?1">
8    </script>
9  </head>
10 <body>
11   <p>Nombre: <input type="text" id="test" value=""></p>
12   <button id="btn1">Mostrar HTML</button>
13 </body>
14 </html>
```





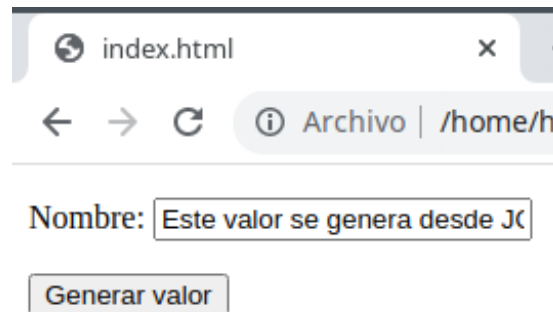
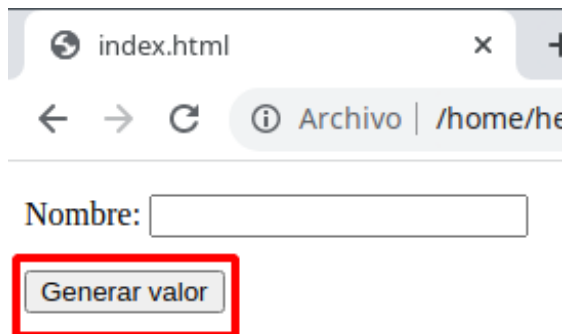
Manipulación del DOM con JQuery (Continuación...)

El método val()

Ejemplo: definiendo el valor del elemento INPUT con **id="test"** “

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("#test").val("Este valor se genera desde JQuery");
4   });
5 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/
7     jquery.min.js"></script>
8   <script src="js/index.js?1">
9 </script>
10 </head>
11 <body>
12   <p>Nombre: <input type="text" id="test" value=""></p>
13   <button id="btn1">Generar valor</button>
14 </body>
15 </html>
```





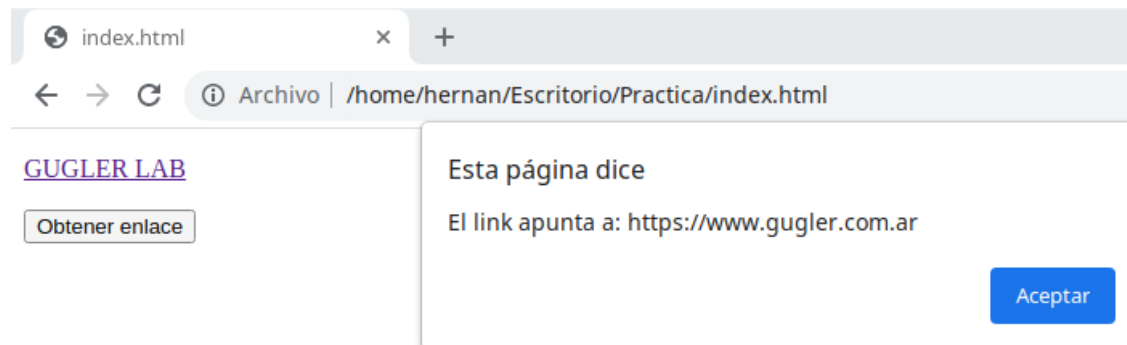
Manipulación del DOM con JQuery (Continuación...)

El método attr()

Este método se utiliza para definir y obtener los valores de los atributos de los elementos del DOM. En el siguiente ejemplo veremos cómo obtener el valor del atributo **href** en un elemento de enlace:

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     let link = $("#link").attr("href");
4     alert("El link apunta a: " + link);
5   });
6 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/
7     jquery.min.js"></script>
8   <script src="js/index.js?1">
9 </script>
10 </head>
11 <body>
12   <p><a href="https://www.gugler.com.ar" id="link">GUGLER LAB</a></p>
13   <button id="btn1">Obtener enlace</button>
14 </body>
15 </html>
```





Manipulación del DOM con JQuery (Continuación...)

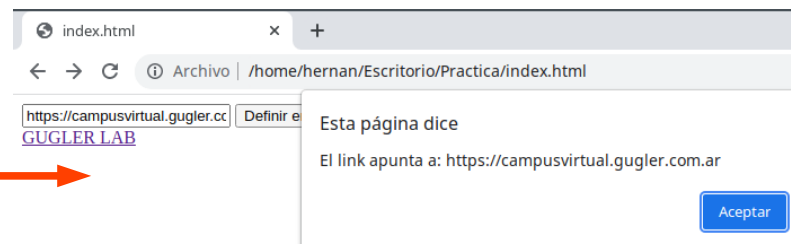
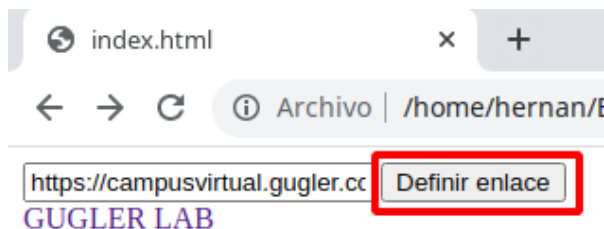
El método attr()

En el caso de querer definir o modificar el valor de un atributo de un elemento del DOM, el método recibirá dos parámetros: primero el atributo a definir o modificar y como segundo parámetro el valor que tendrá el mismo.

`$(selector).attr(atributo,valor);`

```
js > JS index.js > ...
1  $(document).ready(function(){
2      $("#btn1").click(function(){
3          let link = $("#enlace").val();
4
5          $("#link").attr('href',link);
6          alert("El link apunta a: " + link);
7      });
8  });
```

```
<? index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5
6      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/
7      jquery.min.js"></script>
8      <script src="js/index.js?1">
9  </script>
10 </head>
11 <body>
12     <input type="text" id="enlace" value=""/>
13     <button id="btn1">Definir enlace</button>
14     <br>
15     <a href="" id="link">GUGLER LAB</a>
16 </body>
17 </html>
```





~~Manipulación del DOM con JQuery~~

Agregar y eliminar elementos del DOM con JQuery

Manipulando clases CSS con JQuery

Introducción a AJAX con JQuery





Agregar elementos al DOM con jQuery

jQuery también posee métodos para crear o añadir elementos del DOM. En este caso los métodos permiten añadir contenido textual y también nuevos elementos HTML.

A continuación vemos cada uno de ellos:

- **append()** → inserta el contenido al final del elemento seleccionado.
- **prepend()** → inserta el contenido al principio del elemento seleccionado.
- **after()** → inserta el contenido después del elemento seleccionado.
- **before()** → inserta el contenido antes del elemento seleccionado.

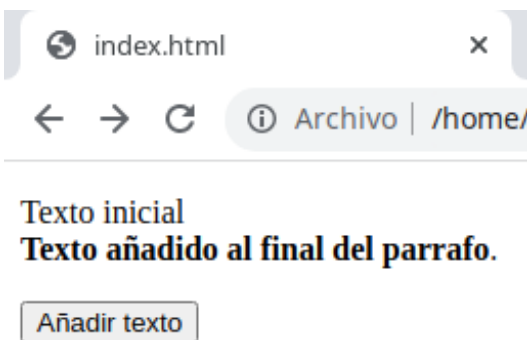
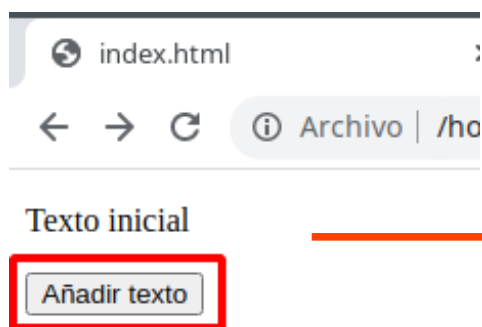


Agregar elementos al DOM con jQuery (Continuación ...)

Añadiendo contenido al final del elemento HTML con **append()**.

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("p").append(" <b>Texto añadido al final del parrafo</b>.");
4   });
5 });
```

```
< index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/
7     jquery.min.js"></script>
8   <script src="js/index.js?1">
9 </script>
10 </head>
11 <body>
12   <p>Texto inicial</p>
13   <button id="btn1">Añadir texto</button>
14 </body>
15 </html>
```



El contenido se añade al final del elemento de párrafo

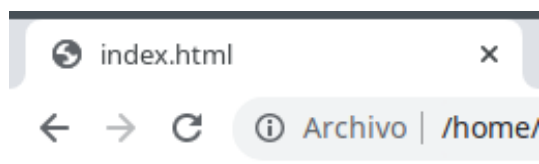


Agregar elementos al DOM con jQuery (Continuación ...)

Añadiendo contenido al inicio del elemento HTML con **prepend()**

```
js > JS index.js > ready() callback > click() callback
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     $("p").prepend("Texto al inicio con <strong>prepend()</strong>. ");
4   });
5 });
```

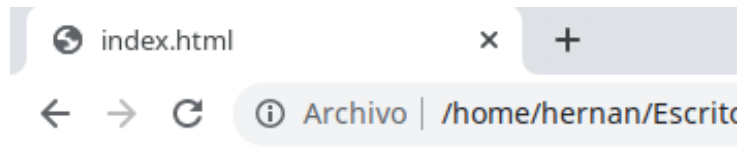
```
<? index.html > html > body > button#btn1
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1"></script>
8 </head>
9 <body>
10  <p>Este es un parrafo</p>
11  <p>Este es otro parrafo</p>
12  <button id="btn1">Añadir texto</button>
13 </body>
14 </html>
```



Este es un parrafo

Este es otro parrafo

Añadir texto



Texto al inicio con **prepend()**. Este es un parrafo

Texto al inicio con **prepend()**. Este es otro parrafo

Añadir texto





Agregar elementos al DOM con jQuery (Continuación ...)

Tanto los métodos **append()** como **prepend()** pueden tomar un numero infinito de parámetros.

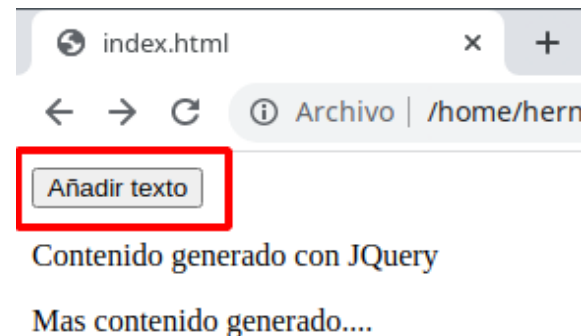
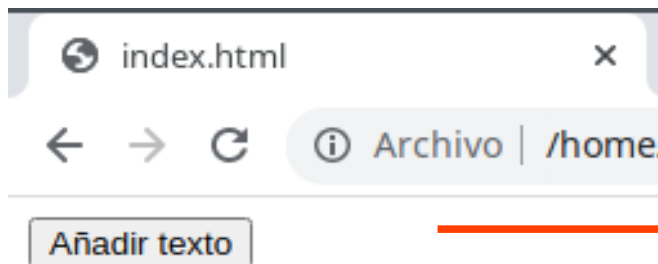
Los nuevos elementos pueden ser generados con texto/HTML (como vimos en los ejemplos anteriores).

En el siguiente ejemplo creamos varios elementos nuevos con el método **append()**.

```
js > JS index.js > ...
```

```
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     let parrafo1 = $("<p></p>").text("Contenido generado con JQuery");
4     let parrafo2 = $("<p></p>").text("Mas contenido generado...");
5     $("body").append(parrafo1, parrafo2);
6   });
7 });
```

```
<? index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1">
8 </script>
9 </head>
10 <body>
11   <button id="btn1">Añadir texto</button>
12 </body>
13 </html>
```



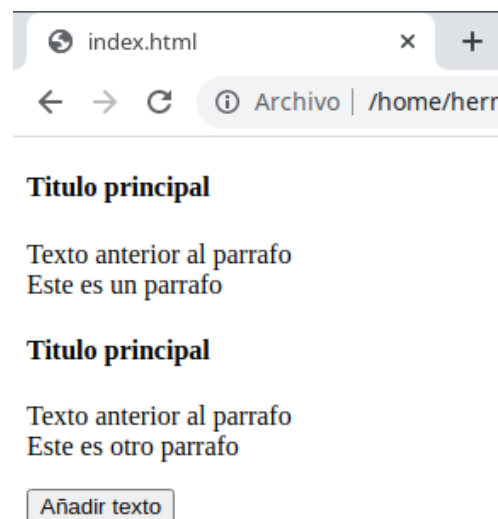
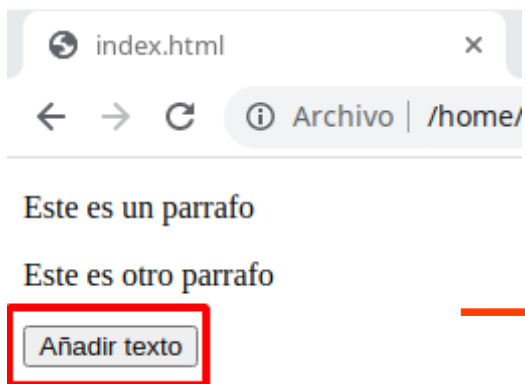


Agregar elementos al DOM con jQuery (Continuación ...)

En este caso creamos varios elementos con el método **prepend()**

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1"></script>
8 </head>
9 <body>
10  <p>Este es un parrafo</p>
11  <p>Este es otro parrafo</p>
12  <button id="btn1">Añadir texto</button>
13 </body>
14 </html>
```

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $("#btn1").click(function(){
3     let titulo = $("<h4></h4>").text("Titulo principal");
4     let texto = $("<span></span>").html("Texto anterior al parrafo <br/>");
5     $("p").prepend(titulo,texto);
6   });
7 });
```



Se crean los elementos **<h4>** y ****, y se añaden al inicio de los elementos párrafo

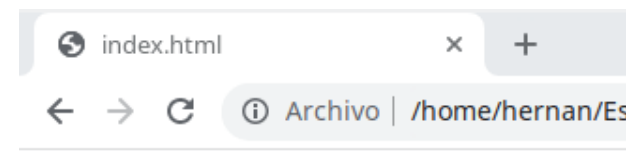
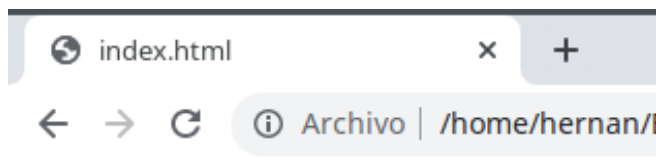
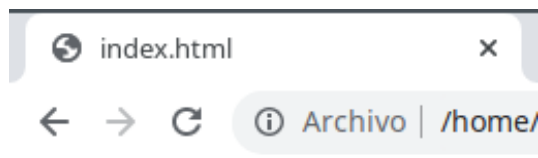


Agregar elementos al DOM con jQuery (Continuación ...)

Añadiendo contenido después del elemento HTML con **after()**

```
js > JS index.js > ...
1 $(document).ready(function(){
2   let contador = 1;
3   $("#btn1").click(function(){
4     let nuevoParrafo = $("
```

```
<? index.html > <? html > <? body > <? h4
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></
7   script>
8   <script src="js/index.js?1">
9   </script>
10 </head>
11 <body>
12   <h4>Titulo principal</h4>
13   <button id="btn1">Añadir Parrafo</button>
14 </body>
15 </html>
```



Titulo principal

Añadir Parrafo

Titulo principal

Parrafo 1 insertado despues del titulo principal

Añadir Parrafo

Titulo principal

Parrafo 2 insertado despues del titulo principal

Parrafo 1 insertado despues del titulo principal

Añadir Parrafo

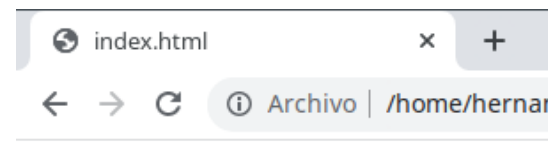
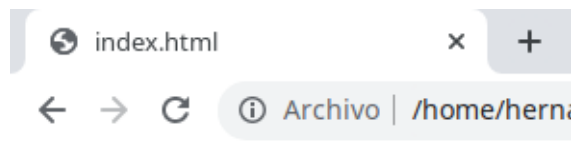
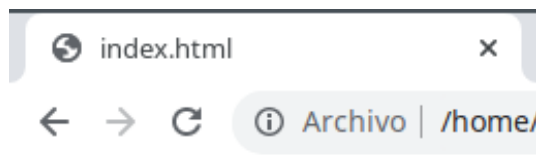


Agregar elementos al DOM con jQuery (Continuación ...)

Añadiendo contenido antes del elemento HTML con **before()**

```
js > JS index.js > ...
1 $(document).ready(function(){
2   let contador = 1;
3   $("#btn1").click(function(){
4     let nuevoParrafo = $("
```

```
index.html > html > body > h4
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></
7   script>
8   <script src="js/index.js?1">
9   </script>
10 </head>
11 <body>
12   <h4>Titulo principal</h4>
13   <button id="btn1">Añadir Parrafo</button>
14 </body>
15 </html>
```



Titulo principal

Añadir Parrafo

Parrafo 1 insertado antes del titulo principal

Titulo principal

Añadir Parrafo

Parrafo 1 insertado antes del titulo principal

Parrafo 2 insertado antes del titulo principal

Titulo principal

Añadir Parrafo



Eliminar elementos del DOM

De la misma manera en que podemos agregar nuevos elementos al DOM, existen además métodos JQuery que permiten eliminar contenido.

Ellos son:

- **remove()** → elimina el elemento seleccionado (y sus elementos hijos).

Ejemplo →

```
$('#parrafo').remove();
```

- **empty()** → elimina los elementos hijos del elemento seleccionado.

Ejemplo →

```
$('#div1').empty();
```





Eliminar elementos del DOM (Continuación ...)

El método **remove()** además admite parámetros opcionales, los cuales se refieren a los elementos HTML que se desean eliminar. Los parámetros pueden ser cualquiera de los **selectores JQuery** válidos.

```
$('p').remove('.prueba');
```



*Elimina todos los elementos párrafo que tengan la clase **prueba**.*

```
$('div').remove('.test, .test2, #principal');
```



*Elimina todos los elementos **div** que tengan la clase **test** ó **test2** ó **id=principal**.*





~~Manipulación del DOM con JQuery~~

~~Agregar y eliminar elementos del DOM con JQuery~~

Manipulando clases CSS con JQuery

Introducción a AJAX con JQuery





Manipulando clases CSS con JQuery

JQuery posee varios métodos para manipular los estilos de los elementos del DOM.

En este caso contamos con métodos para agregar o eliminar clases CSS de un componente HTML:

- **addClass()** → Añade una o más clases CSS en el elemento seleccionado.
- **removeClass()** → Elimina una o más clases CSS en el elemento seleccionado.
- **css()** → Permite devolver o definir un atributo CSS sobre el elemento seleccionado

A continuación vemos un ejemplo de cada uno de ellos





Manipulando clases CSS con JQuery (Continuación ...)

JQuery posee varios métodos para manipular los estilos de los elementos del DOM. Con ellos podremos agregar o eliminar clases CSS o definir y obtener el valor de los atributos CSS del elemento HTML.

- **addClass()** → Añade una o más clases CSS en el elemento seleccionado.

```
$(sel).addClass(nombreClase1 nombreClase2 ...)
```

- **removeClass()** → Elimina una o más clases CSS en el elemento seleccionado.

```
$(sel).removeClass(nombreClase1 nombreClase2 ...)
```

- **css()** → Permite devolver o definir el valor de un atributo CSS sobre el elemento seleccionado.

```
$(sel).css('propiedad-css')
```

→ devuelve el valor de la propiedad CSS.

```
$(sel).css('propiedad-css','valor')
```

→ define la propiedad CSS y su valor.

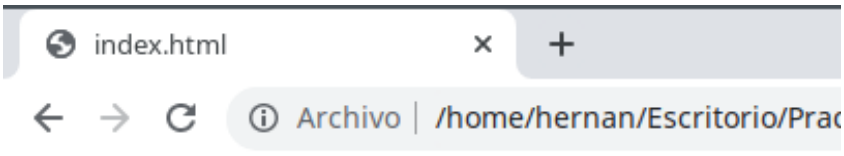


Manipulando clases CSS con JQuery (Continuación ...)

A continuación vemos ejemplo de los métodos `.addClass()`, `.removeClass()` y `.css()`.

Ejemplo agregando clases CSS con `.addClass()`

```
js > JS index.js > ...  
1 $(document).ready(function(){  
2   |   $('p').addClass('importante azul');  
3   | });
```



Este es un parrafo importante

```
<> index.html > ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="utf-8">  
5   <style>  
6     .importante{  
7       font-weight: bold;  
8       font-size: 30px;  
9     }  
10  
11     .azul{  
12       color: blue;  
13     }  
14   </style>  
15   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
16   <script src="js/index.js?1"></script>  
17 </head>  
18 <body>  
19  
20   <p>Este es un parrafo importante</p>  
21 </body>  
22 </html>
```

*Es importante en este caso que las hojas de estilo CSS importadas o clases CSS definidas en el propio documento se encuentren **antes** que el script JavaScript.*



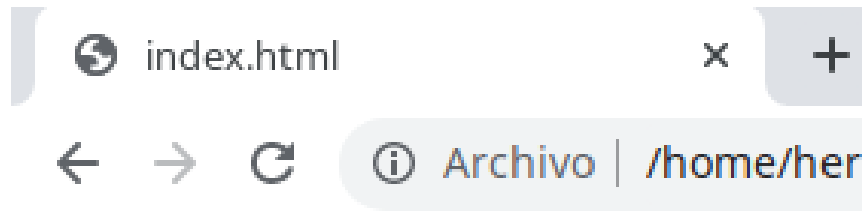


Manipulando clases CSS con JQuery (Continuación ...)

Ejemplo eliminando clases CSS con `.removeClass()`

```
js > JS index.js > ...
```

```
1 $(document).ready(function(){  
2   $('p').removeClass('importante azul');  
3 });
```



Este es un parrafo importante



*Se eliminan las clases **importante** y **azul***

```
<? index.html > ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="utf-8">  
5   <style>  
6     .importante{  
7       font-weight: bold;  
8       font-size: 30px;  
9     }  
10  
11     .azul{  
12       color: blue;  
13     }  
14   </style>  
15   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
16   <script src="js/index.js?1"></script>  
17 </head>  
18 <body>  
19  
20   <p class="importante azul">Este es un parrafo importante</p>  
21 </body>  
22 </html>
```

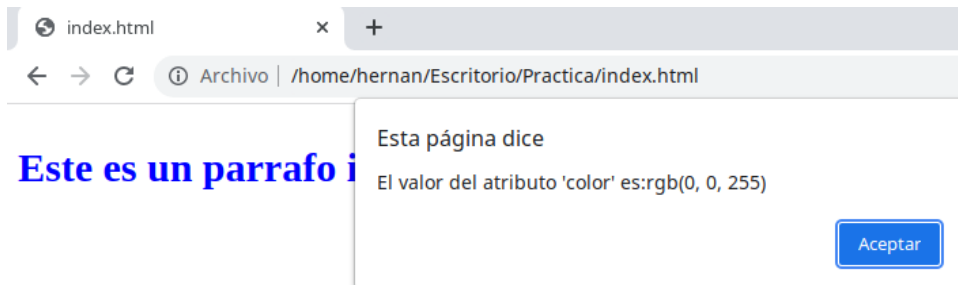




Manipulando clases CSS con JQuery (Continuación ...)

Ejemplo obteniendo el valor de un atributo CSS con `.css()`

```
js > JS index.js > ...
1 $(document).ready(function(){
2   let color = $('p').css('color');
3   alert("El valor del atributo 'color' es:"+color);
4 });
```



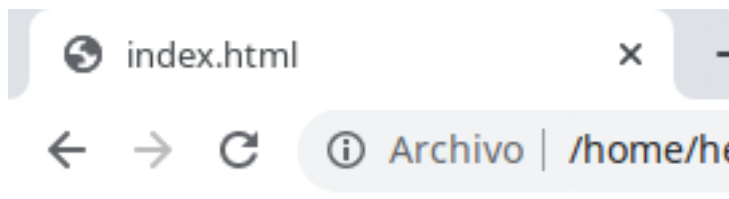
```
<? index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <style>
6     .importante{
7       font-weight: bold;
8       font-size: 30px;
9     }
10
11     .azul{
12       color: blue;
13     }
14   </style>
15   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
16   <script src="js/index.js?1"></script>
17 </head>
18 <body>
19
20   <p class="importante azul">Este es un parrafo importante</p>
21 </body>
22 </html>
```



Manipulando clases CSS con JQuery (Continuación ...)

Ejemplo definiendo un atributo CSS y su valor con `.css()`

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $('p').css('color','red');
3   $('p').css('font-size','20px');
4 });
```



Este es un parrafo importante

```
<? index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <script src="js/index.js?1"></script>
8 </head>
9 <body>
10  <p>Este es un parrafo importante</p>
11 </body>
12 </html>
```

Si queremos definir varios atributos en una misma función podemos pasar un objeto `{ }` con todos los atributos CSS.



```
$('p').css( { 'background-color' : 'red', 'font-size' : '15px' } );
```





~~Manipulación del DOM con JQuery~~

~~Agregar y eliminar elementos del DOM con JQuery~~

~~Manipulando clases CSS con JQuery~~

Introducción a AJAX con JQuery





Introducción a AJAX con JQuery

- AJAX (*Asynchronous JavaScript and XML*) es una característica que permite el intercambio de información entre un cliente y un servidor.
- Permite además actualizar los elementos de una página web sin la necesidad de recargar todo el contenido de la misma.
- JQuery provee métodos que permiten hacer peticiones AJAX en formato texto, HTML, XML o JSON hacia un servidor remoto utilizando las peticiones HTTP GET o POST.

Ejemplos de aplicaciones que utilizan AJAX:

Gmail – Google Maps – Youtube – Facebook tabs





Introducción a AJAX con JQuery (Continuación ...)

Como mencionamos anteriormente, los métodos más comunes para *peticiones-respuesta* entre un cliente y el servidor son los métodos HTTP: GET y POST.

- **GET** → solicita datos de un recurso específico.
- **POST** → envía datos para ser procesados a un recurso específico (aunque también puede ser utilizado para obtener datos del recurso)





Introducción a AJAX con JQuery (Continuación ...)

El método \$.get()

Este método JQuery permite realizar peticiones HTTP GET a un servidor.

\$.get(URL, callback);

Donde

- URL → es la dirección del recurso que se desea consultar
- Callback → parámetro opcional. Es la función que será ejecutada una vez finalizada la petición al recurso.



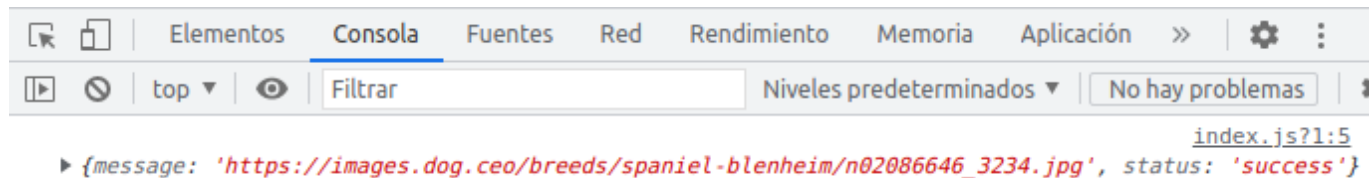


Introducción a AJAX con JQuery (Continuación ...)

El método \$.get()

Veamos un ejemplo a continuación: en este caso utilizamos un servidor API gratuito, el cual obtiene fotos de animales.

```
js > JS index.js > ...  
1  $(document).ready(function(){  
2  
3      //el parámetro DATA de la función recibe la respuesta de la petición  
4      $.get("https://dog.ceo/api/breeds/image/random",function(data){  
5          console.log(data);  
6      });  
7  });
```



Obtenemos el objeto con la respuesta de la solicitud al servidor.



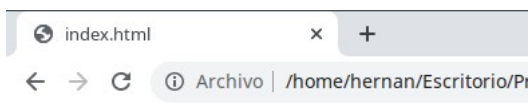
Introducción a AJAX con JQuery (Continuación ...)

El método \$.get()

Al obtener la respuesta del servidor nos permite consultar los valores de la misma y utilizarlos en este caso para generar contenido de forma dinámica en nuestro sitio web.

```
js > JS index.js > ...
1 $(document).ready(function(){
2   $.get("https://dog.ceo/api/breeds/image/random",function(data){
3     /*
4      data = {
5        message: 'https://images.dog.ceo/breeds/spaniel-irish/n02102973_4087.jpg',
6        status: 'success'
7      }
8     */
9     let url = data.message; //obtenemos la url de la respuesta
10
11    /*
12     definimos el atributo src para el elemento de imagen
13     con la URL de respuesta del servidor
14    */
15    $("#imagen").attr('src',url);
16  });
17 });
```

```
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
6   <script src="js/index.js?1"></script>
7 </head>
8 <body>
9
10  <img id="imagen" />
11
12 </body>
13 </html>
```





Introducción a AJAX con JQuery (Continuación ...)

El método \$.post()

Este método JQuery permite realizar peticiones HTTP POST a un servidor.

\$.post(URL, data, callback);

Donde

- URL → es la dirección del recurso que se desea consultar
- Data → parámetro opcional. Especifica datos que serán enviados en la petición.
- Callback → parámetro opcional. Es la función que será ejecutada una vez finalizada la petición al recurso.

Los datos (data) que serán enviados en la petición serán recibidos y analizados por el servidor y luego se devolverá un respuesta.



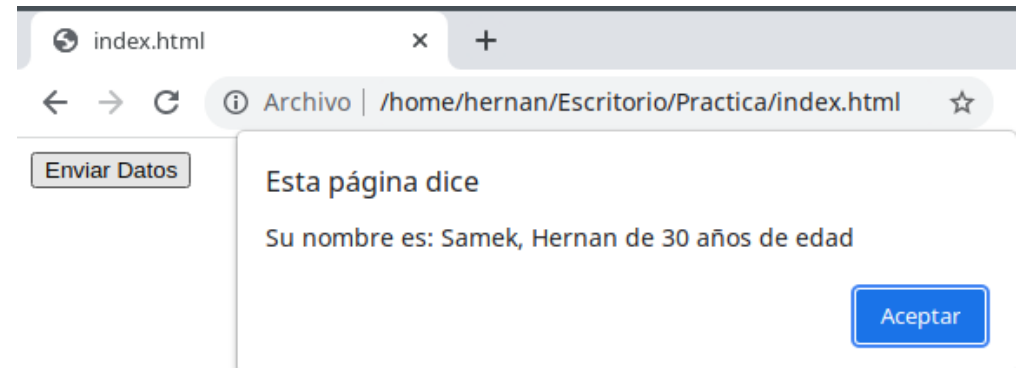


Introducción a AJAX con JQuery (Continuación ...)

El método \$.post()

Veamos un ejemplo a continuación: en este caso utilizamos un servidor PHP local que puede recibir los parámetros **nombre**, **apellido** y **edad**, para procesar la petición y generar una respuesta.

```
js > JS index.js > ...
1  $(document).ready(function(){
2    $("#btn").click(function(){
3      $.post(
4        "http://localhost/clase12/procesaDatos.php",
5        {
6          nombre: 'Hernan',
7          apellido: 'Samek',
8          edad: 30
9        },
10     function(data) {
11       alert(data);
12     }
13   );
14 });
15 });
```



Las peticiones POST van a ser exitosas si los parámetros enviados en la petición coinciden con los esperados por el servidor que las procesa





¿Consultas?

