

Manejo del DOM con jQuery

jQuery HTML



jQuery HTML

jQuery proporciona muchos métodos para manipular elementos de HTML, los más simples pero muy útiles son:

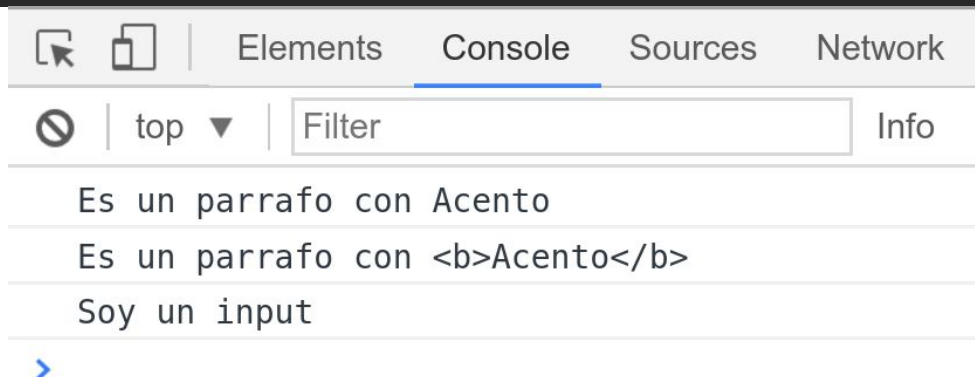
- `text()`
- `html()`
- `val()`

Equivalente a su versión en Javascript puro de

- `textContent`
- `innerHTML`
- `value`

La diferencia es que en jQuery son funciones y no propiedades.

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        console.log($("#p-test").text());
        console.log($("#p-test").html());
        console.log($("#input-test").val());
      });
    </script>
  </head>
  <body>
    <p id="p-test">Es un parrafo con <b>Acento</b></p>
    <input id="input-test" type="text" value="Soy un input"/>
  </body>
</html>
```



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console displays three log messages: 'Es un parrafo con Acento', 'Es un parrafo con Acento', and 'Soy un input'. The first message is the text content, the second is the HTML content (including the bold tag), and the third is the value of the input field.

Elements	Console	Sources	Network
Filter			
Es un parrafo con Acento			
Es un parrafo con Acento			
Soy un input			

jQuery DOM



jQuery DOM

.append(): Inserta contenido al final del elemento de destino que se especifica con el selector.

```
$ ('selector') .append (contenido) ;
```

```
$ ('p') .append ('World! ' ) ;
```

```
<p>Hello </p>
```

.prepend(): Inserta contenido al principio de un elemento especificado.

```
$ ('selector') .prepend ( 'contenido' ) ;
```

jQuery DOM

.before(): Inserta contenido (elementos nuevos o existentes en el DOM) antes del elemento de destino que se especifica mediante el selector.

```
$('selector').before();
```

.after(): Inserta contenido (elementos nuevos o existentes en el DOM) después del elemento de destino que se especifica mediante el selector.

```
$('selector').after();
```

jQuery DOM

.empty(): Elimina los hijos del elemento seleccionado.

```
$('selector').empty();
```

.remove(): Remueve los elementos especificados por el selector.

```
$('selector').remove();
```


.replaceAll(): Sustituye a los elementos seleccionados con nuevos elementos HTML.

```
$('contenido').replaceAll(selector);
```

.wrap(): El método envuelve el elemento HTML especificado (s) alrededor de cada elemento seleccionado.

```
$('button').click(function() {  
    $('p').wrap("<div></div>");  
});
```

Navegando el DOM



.children() y .parent()

.children(): Devuelve un grupo de elementos que contienen todos los hijos inmediatos únicos de cada grupo de elementos previamente seleccionados.

```
$('selector').children();
```

.parent(): Devuelve el padre directo de un elemento. Si la selección es de un grupo de elementos, obtiene un grupo de sus padres directos únicos.

```
$('selector').parent();
```

.parents()

.parents(): Obtiene los ancestros/antepasados de los elementos filtrados por el selector.

Los métodos **parents()** y **parent()** son similares, excepto que este último sólo se desplaza un solo nivel en el árbol DOM.

Además, **\$("html").parent()** devuelve un conjunto que contiene el documento, mientras que **\$("html").parents()** devuelve un conjunto vacío.

```
$( 'selector' ).parents( selector );
```

.find() y .each(),



.find(): Devuelve una nueva selección que contiene aquellos elementos descendientes de la selección previa que cumplen la condición dada.

```
$('selector').find(selector para encontrar);
```

.each(): Itera sobre cada elemento especificado (especificados utilizando el selector) y ejecuta la función de devolución de llamada para cada elemento.

```
$('selector').each(function);
```

.first(), y .next(),



.first(): Devuelve el primer elemento en el conjunto de los elementos coincidentes.

```
$('selector').first();
```

.next(): Devuelve el hermano inmediatamente siguiente del elemento en el conjunto de los elementos coincidentes.

```
$('selector').next();
```

.prev() y .siblings()

.prev(): Devuelve el hermano anterior del elemento en el conjunto de los elementos coincidentes.

```
$('selector').prev();
```

.siblings(): Devuelve todos los hermanos del elemento especificado.

```
$('selector').siblings();
```

jQuery CSS



.addClass(): Agrega una clase a la lista de clases del elemento seleccionado.

```
$('selector').addClass('my-class');
```

.removeClass(): Elimina la clase que se pasa como parámetro de la lista de clases del elemento seleccionado.

```
$('selector').removeClass('my-class')  
;
```

.toggleClass(): Agrega o elimina la clase dependiendo si esta existe o no en el elemento seleccionado

```
$('selector').toggleClass('my-class');
```

.css(): Obtiene o setea el valor de una propiedad de CSS.

```
$('selector').css('background-color');
```

```
$('selector').css('background-color', 'yellow');
```

jQuery CSS



.width(): Obtiene o setea el ancho del elemento excluyendo padding, border, margin.

```
$('selector').width();
```

.height(): Obtiene o setea el alto del elemento excluyendo padding, border, margin.

```
$('selector').height();
```

.innerWidth(): Obtiene el ancho de un elemento excluyendo margin, border

```
$('selector').innerWidth();
```

.innerHeight(): Obtiene el alto de un elemento excluyendo margin, border

```
$('selector').innerHeight();
```

.outerWidth(): Obtiene el ancho de un elemento excluyendo margin a menos que se le pase true con parámetro

```
$('selector').outerWidth();
```

.outerHeight(): Obtiene el alto de un elemento excluyendo margin a menos que se le pase true con parámetro

```
$('selector').outerHeight();
```