



Juan Diego Pérez Jiménez  
@pekechis  
<https://github.com/pekechis>

# ÍNDICE

- Introducción
- `.empty()` / `.html()` / `.text()`
- `.append()` / `.prepend()` / `.appendTo()` / `.prependTo()`
- `.wrap()` / `.unwrap()` / `.wrapAll()` / `.wrapInner()`
- `.val()` -> Sí....aquí

# 1. Introducción



**Hasta ahora** hemos “respetado” el **árbol HTML**,  
no hemos tocado el **DOM** y nos hemos limitado  
a **modificar estilos y atributos**.

# 1. Introducción



**Pero...en ocasiones querremos **modificar esa estructura, añadiendo y borrando** elementos o bien **modificando el contenido** de los mismos. **jQuery** tiene muchas funciones para eso.**

## 2. .empty() / .html() / .text()

```
$(function() {
```

```
    //Borra todos los nodos hijos (y su contenido) de los elementos
```

```
    // seleccionados
```

```
    $(".article li:nth-child(3)").empty();
```

```
    //Obtiene el HTML del PRIMER ELEMENTO de los seleccionados
```

```
    //Es decir TODO lo que va entre la apertura y el cierre de ese elemento
```

```
    var content = $("li").html();
```

```
});
```



## 2. .empty() / .html() / .text()

```
$(function() {
```

```
    //SUSTITUYE el contenido de todos los elementos seleccionados
```

```
    $("ul").html("<li>UNO</li>");
```

```
    //Tengo disponible posición y antiguo texto si le paso una función
```

```
    $("ul").html(function(index,oldText) {
```

```
        .....
```

```
    });
```


```
    //El uso de .text() es análogo a .html() pero con la diferencia de que trata
```


```
    //TODO como TEXTO y obvia las etiquetas
```

```
});
```

### 3. .append() / .prepend() / .appendTo() / .prependTo()

**.append()** y **.prepend()** son **análogos**, sirve para **añadir nuevos elementos HTML** a nuestro árbol.

**.append("some\_content")**  JUSTO AL **FINAL**

**.prepend("some\_content")**  JUSTO AL **PRINCIPIO**

### 3. .append() / .prepend() / .appendTo() / .prependTo()

```
<ul>  
  <li>UNO</li>  
  <li>DOS</li>  
</ul>
```

`$("li").append("<li>TRES</li>");`

```
<ul>  
  <li>UNO</li>  
  <li>DOS</li>  
  <li>TRES</li>  
</ul>
```

`$("li").prepend("<li>TRES</li>");`

```
<ul>  
  <li>TRES</li>  
  <li>UNO</li>  
  <li>DOS</li>  
</ul>
```



### 3. .append() / .prepend() / .appendTo() / .prependTo()

**.appendTo()** y **.prependTo()** son análogos,  
sirve para **añadir nuevos elementos HTML** a  
**OTROS** elementos HTML.

**.appendTo("some\_selector")**       JUSTO AL **FINAL**

**.prependTo("some\_selector")**       JUSTO AL **PRINCIPIO**

### 3. .append() / .prepend() / .appendTo() / .prependTo()

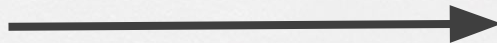
<ul>

<li>UNO</li>

<li>DOS</li>

</ul>

`$("<li>TRES</li>").appendTo("<li>");`



<ul>

<li>UNO</li>

<li>DOS</li>

<li>TRES</li>

</ul>



`$("<li>TRES</li>").prependTo("<li>");`



<ul>

<li>TRES</li>

<li>UNO</li>

<li>DOS</li>

</ul>



## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$(function() {
```

```
//WRAP - Añade cierta estructura HTML alrededor de todos los
```

```
//elementos seleccionados
```

```
// seleccionados
```

```
$(“article”).wrap("<div class=“article_outer”></div>”);
```

```
//O usando una función. La posición y $(this) disponibles
```

```
});
```

## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$(function() {
```

```
    //WRAP - Añade cierta estructura HTML alrededor de todos los elementos seleccionados  
    // seleccionados
```

```
    $("article").wrap("<div class='article_outer'></div>");
```

```
    //O usando una función. La posición y $(this) disponibles
```

```
    //El valor de retorno será la estructura envolvente
```

```
    $("article").wrap(function(index) {
```

```
        ....
```

```
    });
```

```
});
```



## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$("article").wrap("<div class='article_outer'></div>");
```



## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$(function() {
```

```
    //UNWRAP - Elimina el padre de los elementos seleccionados
```

```
    //Deshacer el cambio anterior hecho con WRAP
```

```
    $("article").unwrap();
```

```
    $("article").unwrap(".article_outer"); //Si queremos comprobar que el padre cumple condiciones
```


```
    //WRAPALL() es ANÁLOGO a WRAP pero sólo AÑADE sólo un elemento envolvente que cubre todos los seleccionados. CUIDADO CON ELEMENTOS "INTRUSOS"
```

```
    //WRAPINNER() es ANÁLOGO a WRAP pero el elemento envolvente se añade no como padre sino al contenido de los seleccionados.
```

```
});
```

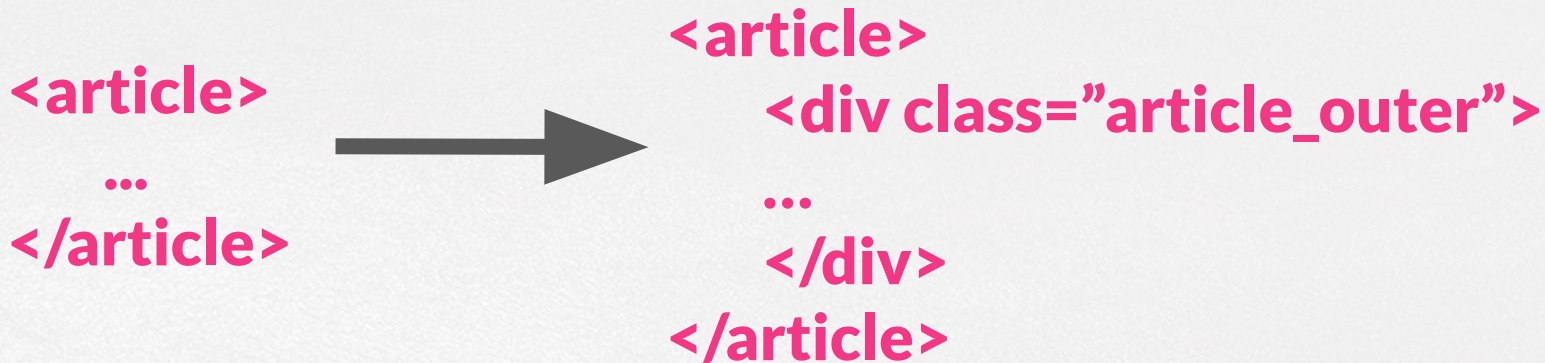
## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$(“article”).wrapAll(“<div class=“article_outter”></div>”);
```

<pre><i>&lt;article&gt;. .&lt;/article &gt; &lt;article&gt; &lt;/article&gt;</i></pre>		<pre><i>&lt;div class=“article_outter”&gt;   &lt;article&gt;   &lt;/article&gt;   &lt;article&gt;   &lt;/article&gt; &lt;/div&gt;</i></pre>
--	---	---

## 4. .wrap() / .unwrap() / .wrapAll() / .wrapInner()

```
$(“article”).wrapInner("<div class='article_inner'></div>”);
```





## 5. .val()

(obtener y fijar campos de formulario)

`$(function() {`

*//Obtener el valor de un campo de formulario. El primero de los seleccionados*

`var valor = $("some_selector").val();`

*//Fijar el valor de un campo del formulario*

`$("some_selector").val("some_value");`

*//O usando una función. La posición, el valor actual y \$(this) disponibles*

*//El valor de retorno será el valor del elemento seleccionado*

`$("some_selector").val(funcion(index,valor) {`

`....`

`});`

`});`

## 5. `.val()`

(obtener y fijar campos de formulario)

```
$(function() {
```

```
    var valor = $("input[type=text]").val();
```

```
    $("input[type=text]").val("Insert Value");
```

```
    $("input[type=text]").val(function(index,valor) {  
        return valor+"->" +index;  
    });
```

```
});
```

# THANKS!

## Any questions?

You can find me at @pekechis &  
<https://github.com/pekechis>