

Acceder y modificar atributos HTML

- **Lectura de un atributo:** usamos `attr()` para recuperar el valor de un atributo, debe recibir una cadena con el nombre del atributo que queremos recuperar.

```
$("#campotexto").attr("value")
```

Si invocamos `attr` sobre un objeto jQuery con varios elementos, devolverá el atributo del primero de los elementos.

Si el elemento no tiene definido el atributo devolvería `undefined`.

```
<a href="http://www.elpais.com" title="El País">El País</a> <br>  
<a href="http://www.mozilla.org" title="Mozilla">Mozilla Foundation</a> <br>  
<a href="http://es.openoffice.org/" title="OpenOffice">Open Office</a>  
$("a").attr("title")
```

Devuelve el "title" del primer elemento, hay que hacer un recorrido con `each` para acceder a cada uno de los enlaces.

```
$("a").each(function(i){  
    var titulo = $(this).attr("title");  
    alert("Atributo title del enlace " + i + ": " + titulo);  
});
```

Acceder y modificar atributos HTML

- **Modificar un atributo:** se hace cuando attr() recibe dos cadenas de texto, la primera con el nombre del atributo y la segunda con el nuevo valor a asignar:

```
$('li').attr("type", "square");
```

- **Modificar varios valores de atributos a la vez:** Se hace pasando un objeto con pares atributo/valor, si esos atributos no existían, simplemente los crea, hay que crear variables con notación objeto

```
$('a').attr({  
    'title': 'Title modificado por jQuery',  
    'href': 'http://www.desarrolloweb.com',  
    'style': 'color: #f80'  
});
```

- **Asignar un valor procesado por una función:** Podemos usar una función para obtener el valor para asignar a un atributo. Dos parámetros: el primero con el nombre del atributo y el segundo la función que debe devolver el valor a asignar
`$('input.fecha').attr("value", function(indiceArray){ //indiceArray es el índice del elemento
 var f = new Date();
 return f.getDate() + "/" + (f.getMonth() +1) + "/" + f.getFullYear();
});`

Acceder y modificar atributos HTML

Eliminar un atributo de uno o varios elementos: para borrar un atributo se usa `removeAttr()`, recibe el nombre del atributo a eliminar y lo borra del elemento. Se borra por completo de la etiqueta tomando el valor por defecto que tenga configurado el navegador.

```
<html> <head>
  <script src="../../jquery-X.X.X.min.js" type="text/javascript"></script>
  <script>
    $(document).ready(function(){
      $("#boton").click(function(i){
        $("td").removeAttr("noWrap"); //"noWrap" tiene que estar con la "W"
        mayúscula para que funcione en Iexplorer.
      });
    });
  </script>
</head> <body>
  <table width="50"> <tr> <td nowrap>
    Con nowrap, todo el texto se muestra en la misma línea! la tabla mide 50 pixeles de
    ancho,tendrían varias líneas!
  </td> </tr> </table>
  <input type="Button" id="boton" value="Quitar nowrap">
</body> </html>
```

prop() y diferencias con attr()

- El método prop() disponible desde jQuery 1.6 sirve para acceder y modificar propiedades de elementos y attr() para atributos.
- Además, tenemos el método val() que sirve específicamente para acceder y modificar el atributo value. Es el que deberíamos usar siempre que queremos ver el valor que haya cargado en "value", o modificarlo.
- **Atributo**: cualquier cosa que tengamos en una etiqueta HTML para personalizarla.

`...` //"style" es un atributo de la etiqueta HTML.

- **Propiedad**: cualquier cosa a la que podamos acceder desde una propiedad de un objeto Javascript.

`document.forms[0].elements[0].checked`

Uso del método prop()

- Sirve para modificar propiedades nativas de Javascript de los elementos de una página.
- Con único parámetro permite acceder al valor de la propiedad que indiquemos.
- Con dos parámetros, la propiedad a modificar y el valor que queremos introducir.

`$(elemento).prop("checked");` //devuelve un booleano que indica si un checkbox está o no marcado.

- Si quisiéramos modificar el estado del checkbox, hacemos lo siguiente:

`$(elemento).prop("checked", true);`

Cuándo utilizar prop() y cuándo usar attr()

- attr() solo dará el valor de atributos HTML

Atributos que se modifican con attr(): class, id, href, label, src, title...

- prop() dará el valor de las propiedades del DOM de Javascript.

Propiedades que se modifican con prop(): autofocus, checked, async, multiple, readOnly...

- Todas las propiedades booleanas de los objetos del DOM se tienen que acceder por prop() y se han desactivado en attr().

Método `css()` de jQuery

- **`.css(nombre_propiedad_css)`** con un solo parámetro se recibe el valor de una propiedad CSS.

```
<div id="micapa" style="color: red;">hola!</div>  
$("#micapa").css("color");
```

- **`.css(nombre_propiedad_css, valor)`** con el nombre de una propiedad CSS y un valor se asigna un nuevo estado a dicho atributo.
 - Cambiar un único atributo: `$("#micapa").css("color", "green");`
 - Cambiar varios atributos: notación de objeto.

```
$("#micapa").css({  
  "background-color": "#ff8800",  
  "position": "absolute",  
  "width": "100px",  
  "top": "100px",  
  "left": "200px"  
});
```

```
$("#micapa").mouseover(function(){  
  antiguoLeft = parseInt($(this).css("left"));  
  $(this).css("left", antiguoLeft + 10 + "px");  
});
```

Método `css()` de jQuery

- Cambiar un atributo con el valor de una función: Pasar una función que tenga un return, esa función recibe dos valores. El índice del elemento jQuery y el valor actual del atributo a cambiar.

```
$("#micapa").click(function(){  
    $(this).css("width", function(index, value){  
        var aumento = prompt("cuanto quieres aumentar?", "25");  
        return (parseInt(value) + parseInt(aumento)) + "px";  
    });  
})
```

///la función devuelve un valor, que es lo que se colocará en el atributo width.

Funciones CSS para el tamaño y posición de elementos

- **innerWidth() e innerHeight():** Reciben un jQuery, retornan las dimensiones internas del 1er elemento, anchura y altura contando el padding pero no el borde.
- **outerWidth() e outerHeight():** Reciben objeto jQuery, retornan las dimensiones externas del 1er elemento, anchura y altura contando padding y borde.

Si no hay borde inner e outer son los mismos.

```
function dimensionCapa(capa){  
  capa = $(capa);  
  var dimensiones = "Dimensiones internas: " + capa.innerWidth() + "x" + capa.innerHeight();  
  dimensiones += "\nDimensiones externas: " + capa.outerWidth() + "x" + capa.outerHeight();  
  alert(dimensiones);  
}
```

- **offset() y position():** Posición en la página. Reciben objeto jQuery retornan la posición del 1er elemento en un objeto con dos atributos, "top" y "left", con los píxeles respecto la esquina superior izquierda. offset() indica la posición real, contando los márgenes. position() indica la posición si no tuviera márgenes.

```
posicionReal = $("#idelemento").offset();  
alert(posicionReal.top);  
alert(posicionReal.left);
```

Exercicis de la quinta part!!!!!!!