

Comprobar sin un checkbox está seleccionado con jQuery jQuery

😊 Juan Padial 📅 30 junio, 2014 - Actualizado el 9 octubre, 2015

Para comprobar con jQuery si un checkbox está seleccionado podemos utilizar varios métodos. En la mayoría de situaciones cualquiera de las vías tiene el mismo resultado y la verdad es que no sabría decirte cuál es mejor sin ver el contexto exacto. En cualquier caso, **se trata de comprobar el atributo o propiedad `checked` del elemento checkbox**. Veamos como hacerlo.

1 Utilizando el método `.prop()`

Con el método `.prop()` podemos obtener el valor de la propiedad `checked`, que recordemos es un valor lógico (`true` / `false`).

```
if( $('.micheckbox').prop('checked') ) {
    alert('Seleccionado');
}
```

El método `.prop()` está disponible desde jQuery 1.6, en versiones anteriores hay que utilizar el método `.attr()`:

```
if( $('.micheckbox').attr('checked') ) {
    alert('Seleccionado');
}
```

Propiedades y atributos no son exactamente lo mismo. En la mayoría de los casos es preferible el uso de `.prop()`, ya que **refleja el estado actual del objeto DOM** que representa al elemento HTML con el que estamos trabajando. En [esta entrada de StackOverflow](#) explican muy bien la diferencia entre propiedad y atributo.

2 Utilizando el método `.is(selector)`

El método `.is(selector)` permite comprobar si un elemento cumple con el selector especificado. Utilizándolo con el **pseudo selector** `:checked` podremos comprobar si el checkbox se encuentra seleccionado.

```
if( $('.micheckbox').is(':checked') ) {
    alert('Seleccionado');
}
```

El método `.is()` es **especialmente útil dentro de callbacks** ya que nos permite comprobar el estado del objeto del callback sin modificarlo y sin crear objetos nuevos (vea un ejemplo de este uso un poco más abajo, en el ejemplo de captura del evento de selección).

3 Obtener todos los checkboxes seleccionados

Para obtener todos los elementos checkbox que se encuentran seleccionados podemos utilizar el pseudo selector `:checked`. Por ejemplo:

```
$('.micheckbox:checked').each(
    function() {
        alert("El checkbox con valor " + $(this).val() + " está seleccionado");
    }
);
```

El código anterior obtendría todos los elementos `.micheckbox` (clase "micheckbox") que se encuentren seleccionados, pero no específicamente elementos tipo `checkbox` (el selector `:checked` en jQuery coincide tanto con los elementos `checked` como con los elementos `selected`). Podríamos limitarlo sólo a elementos checkbox, por ejemplo, del siguiente modo:

```
$("input:checkbox:checked").each(
    function() {
        alert("El checkbox con valor " + $(this).val() + " está seleccionado");
    }
);
```

4 Capturar el evento de selección del checkbox

Hasta ahora hemos comprobado si el checkbox está seleccionado. Ahora, si queremos hacer algo justo cuándo el checkbox es seleccionado, tenemos que capturar ese momento. Hay muchos tutoriales que proponen el uso del evento `click`. Por ejemplo:

```
$( '.micheckbox' ).on( 'click', function() {
    if( $(this).is(':checked') ){
        // Hacer algo si el checkbox ha sido seleccionado
        alert("El checkbox con valor " + $(this).val() + " ha sido seleccionado");
    } else {
        // Hacer algo si el checkbox ha sido deseleccionado
        alert("El checkbox con valor " + $(this).val() + " ha sido deseleccionado");
    }
});
```

Sin embargo, **creo que es mejor utilizar el evento** `change` ya que es lanzado específicamente cuándo el checkbox cambia de estado, que es lo que buscamos, independientemente de si ha sido "clickeado" o no. Un ejemplo básico tendría este pinta:

```
$(".micheckbox").on( 'change', function() {
    if( $(this).is(':checked') ) {
        // Hacer algo si el checkbox ha sido seleccionado
        alert("El checkbox con valor " + $(this).val() + " ha sido seleccionado");
    } else {
        // Hacer algo si el checkbox ha sido deseleccionado
        alert("El checkbox con valor " + $(this).val() + " ha sido deseleccionado");
    }
});
```