

Join the Stack Overflow Community

Stack Overflow is a community of 6.6 million programmers, just like you, helping each other. Join them; it only takes a minute:

[Sign up](#)

How to make first option of <select> selected with jQuery?

```
<select id="target">
  <option value="1">...</option>
  <option value="2">...</option>
</select>
```

[jquery](#) [select](#)

asked Sep 12 '09 at 4:28

[omg](#)

36.4k 108 238 318

The first solution that worked for me was the 5th answer by Esben – [Casey Crookston](#) Jan 6 at 19:53

22 Answers

```
$("#target").val($("#target option:first").val());
```

answered Sep 12 '09 at 4:31

[David Andres](#)

20.5k 5 35 34

231 Should be noted that this more easily done with `$("#target")[0].selectedIndex = 0;` – [David Andres](#) Sep 12 '09 at 4:34

6 It works, but it's based on 1st element *value*, if 1st element contains empty value, it will select nearest element WITH value – [evilReiko](#) Mar 1 '11 at 12:37

1 Seems to work great, except for IE6. Unsure of 7-8, works on 9. – [Nicholi](#) Feb 3 '12 at 21:53

2 I recant my previous statement, was getting ahead of myself. `$("#target").val("option:first");` works most everywhere but IE6 `$("#target").val($("#target option:first").val());` will work in IE6, because you are literally looking up the first value, and entering it as target's value. Two id lookups instead of one. – [Nicholi](#) Feb 4 '12 at 1:56

1 And it's actually necessary for IE6, 7, and 8. Whereas `.val("option:first")` works in IE9 (and Safari and Firefox and Chrome and Opera). – [Nicholi](#) Feb 4 '12 at 2:20

More jobs
means more choice



stackoverflow
JOBS

[Get started](#)

```
// remove "selected" from any options that might already be selected
$("#target option[selected='selected']").each(
  function() {
    $(this).removeAttr('selected');
  }
);
```

```
// mark the first option as selected
$("#target option:first").attr('selected', 'selected');
```

edited Dec 8 '11 at 21:28

[jmort253](#)

24.3k 7 67 93

answered Sep 12 '09 at 4:34

[user149513](#)

1,243 1 9 11

1 Good second approach. Please add the value of the selected attribute. For example, `attr("selected", "selected")`. Without this extra parameter, the selection isn't made. – [David Andres](#) Sep 12 '09 at 4:36

yes you are correct..i forgot to add that value.. – [user149513](#) Sep 12 '09 at 5:05

1 What if some option is already selected? – [Egor Pavlikhin](#) Sep 29 '10 at 6:20

@EgorPavlikhin - I just edited the answer to include JQuery code to remove the "selected" flag from any options that might have already been selected. Now the answer is correct :) – [jmort253](#) Dec 8 '11 at 21:29

16 There is no need for the `each` function, should be: `$('#target option[selected="selected"]').removeAttr('selected')` also now should be used with `removeProp` and `prop` instead. – [vsync](#) Dec 22 '11 at 9:59

when you use

```
$("#target").val($("#target option:first").val());
```

This will not work in Chrome and Safari if the first option value is null.

I prefer

```
$("#target option:first").attr('selected', 'selected');
```

because it can work in all browsers.

answered Dec 15 '10 at 10:49



[Jimmy Huang](#)

2,903 2 15 21

2 You also need to "unselect" any prior selected elements for this to work in more cases. See the answer from James Lee Baker for details. – [Charleston Software Associates](#) Aug 24 '13 at 14:15

You can try this

```
$("#target").prop("selectedIndex", 0);
```

edited Aug 8 '13 at 9:09



[Romesh](#)

1,551 2 12 36

answered Aug 8 '13 at 8:47



[Esben](#)

467 4 2

3 Hopefully this answer gets more attention. – [Jeff Hines](#) Aug 9 '13 at 17:05

This does not work properly in all cases. I have a parent/child cascading drop downs. Select parent #1, display child #1, select parent #2 display child #2. Any time they select a new parent it should set the corresponding child back to the first option. This solution does not do this. It leaves the child menu "sticky". See the `option:selected.prop('selected',false) / option:first.prop('selected','selected')` solution on this question for the answer that works in more scenarios. – [Charleston Software Associates](#) Aug 24 '13 at 14:08

This also works when you are using `$(this)` – [DalekSall](#) May 24 '16 at 13:20

1 this is much better – [jned29](#) Jun 24 '16 at 3:17

this doesnt trigger onChange event. At least in chrome. – [Tomasz](#) Dec 29 '16 at 9:32

Changing the value of the select input or adjusting the selected attribute can overwrite the default `selectedOptions` property of the DOM element, resulting in an element that may not reset properly in a form that has had the reset event called.

Use jQuery's `prop` method to clear and set the option needed:

```
$("#target option:selected").prop("selected", false);
$("#target option:first").prop("selected", "selected");
```

answered May 16 '13 at 23:12



[James Lee Baker](#)

462 5 9

2 Perfect. I have cascading drop down menus. When a user selects parent option 2 then child option 3 I do not want that to "stick". In other words the user then selects parent #1, then re-selects parent #2. When that happens I want the child menu to reset to the first option (which is "Any" in my case). All the other solutions here fail on Firefox v19 and Chrome on Linux. – [Charleston Software Associates](#) Aug 24 '13 at 14:07

```
$("#target")[0].selectedIndex = 0;
```

edited May 24 '11 at 13:37



[GSerg](#)

49.3k 10 78 132

answered May 23 '11 at 20:34



[danactive](#)

287 3 2

This will not work in cases where a drop down has a pre-selected item. See my comments on the selected answer for details. – [Charleston Software Associates](#) Aug 24 '13 at 14:15

If you have disabled option, you may add **not([disabled])** to prevent selecting them which result into the following:

```
$("#target option:not([disabled]):first").attr('selected','selected')
```

answered Jul 29 '11 at 16:09



[Melanie](#)

666 1 12 35

1 Good point about disabled options. Nice addition to the answers presented! – [Charleston Software Associates](#) Aug 24 '13 at 14:14

One subtle point I *think* I've discovered about the top voted answers is that even though they correctly change the selected value, they do not update the element that the user sees (only when they click the widget will they see a check next to the updated element).

Chaining a `.change()` call to the end will also update the UI widget as well.

```
$("#target").val($("#target option:first").val()).change();
```

(Note that I noticed this while using jQuery Mobile and a box on Chrome desktop, so this may not be the case everywhere).

edited Dec 8 '15 at 22:15



[Marc](#)

5,795 13 40 80

answered Jul 11 '13 at 18:37



[Bradley Bossard](#)

1,378 1 15 23

Another way to reset the values (for multiple selected elements) could be this:

```
$("#selector").each(function(){
    /*Perform any check and validation if needed for each item */

    /*Use "this" to handle the element in javascript or "$(this)" to handle the element
    with jquery */

    this.selectedIndex=0;
});
```

edited Jun 19 '13 at 9:45

[user133408](#)

answered May 16 '12 at 10:21



[Nick Petropoulos](#)

101 1 2

4 +1 it's actually a good answer, but you could have made it more specific to the question domain; instead of `$("#selector")` you could have just written `$('#target')` – [Jack](#) May 17 '12 at 4:12

This is what I needed as I have multiple selects, thanks :) – [Fiona - myaccessible.website](#) Oct 23 '13 at 14:30

I've found that just setting `attr selected` doesn't work if there's already a selected attribute. The code I use now will first unset the selected attribute, then select the first option.

```
$('#target').removeAttr('selected').find('option:first').attr('selected', 'selected');
```

answered Sep 16 '11 at 6:02



[Francis Lewis](#)

4,868 4 32 50

This works well and is very similar to the answer by James Lee Baker noted elsewhere on this question. I like the other answer with `option:selected` and `option:first` as the execution of `option:first` is likely to be faster (compute-wise) than `find()`. – [Charleston Software Associates](#) Aug 24 '13 at 14:13

```
$('#newType option:first').prop('selected', true)
```

answered Aug 19 '13 at 3:46



[FFFFF](#)

86 2 5

This works only if there's no already selected options. See James Lee Baker's answer. – [Webars](#) Jun 13 '14 at 17:13

I know this question has been answered, but here is how I would do it

```
$("#target option")
  .removeAttr('selected')
  .find(':first') //you can also use .find('[value=MyVal]')
  .attr('selected', 'selected');
```

answered Mar 2 '12 at 19:38



Fabrizio

2,124 1 16 26

Works perfectly in a \$.each loop, thanks! – [Martin Gemme](#) Apr 11 '13 at 1:54

- 1 This works well and is very similar to the answer by James Lee Baker noted elsewhere on this question. I like the other answer with option:selected and option:first as the execution of option:first is likely to be faster (compute-wise) than find(). – [Charleston Software Associates](#) Aug 24 '13 at 14:11

Simple like that:

```
$('#target option:first').prop('selected', true);
```

edited Feb 19 '16 at 9:27



Azim

16.1k 6 19 36

answered Feb 19 '16 at 3:59



Ramon Schmidt Rocha

114 4

- 1 this is much better – [ARUN Edathadan](#) Jul 25 '16 at 12:00

Although the each function probably isn't necessary ...

```
$('#select').each(function(){
  $(this).find('option:first').prop('selected', 'selected');
});
```

works for me.

answered Oct 24 '14 at 13:23



Darth Jon

157 4 13

If you going to use the first option as a default like

```
<select>
  <option value="">Please select an option below</option>
  ...
```

Then you can just use.

```
$('#select').val('');
```

Nice and simple.

answered Aug 20 '12 at 0:20



Gary - ITB

29 1

- 2 This will not work in IE. – [Vinothkumar Arputharaj](#) Mar 25 '13 at 14:46

```
USE $("#selectbox
option:first").val()
```

Please find the working simple

[jsfiddle](#)

answered Jul 7 '15 at 19:29



Chetan

779 2 10 21

```
alert($("#target option:first")
).text());
```

answered Aug 11 '16 at 9:49



GobsYats

11 2

```
$("#target").val(null);
```

worked fine in chrome

answered Sep 15 '14 at 7:55



[Krishna Murthy](#)

121 1 5

If you're storing the jQuery object of the select element:

```
var jQuerySelectObject = $("....");  
...  
jQuerySelectObject.val(jQuerySelectObject.children().eq(0).val());
```

edited Nov 26 '14 at 22:14

answered Nov 26 '14 at 21:26



[Daniel K](#)

1,175 2 18 33

On the back of James Lee Baker's reply, I prefer this solution as it removes the reliance on browser support for `:first:selected` ...

```
$('#target').children().prop('selected', false);  
$('#target').children()[0].prop('selected', 'selected');
```

answered Oct 18 '16 at 15:30



[Matthew Scott](#)

1

It only worked for me using a `trigger('change')` at the end, like this:

```
$("#target option:first").attr('selected','selected').trigger('change');
```

answered Nov 10 '16 at 21:19



[Pablo S G Pacheco](#)

1,181 8 18

Check this best approach using **jQuery** with **ES2015**:

```
$('.select').each((i, item) => {  
  var $item = $(item);  
  $item.val($item.find('option:first').val());  
});
```

Regards, Nicholls

answered Dec 20 '16 at 20:09



[Juan David](#)

441 5 13