

Jquery selector input[type=text]

[Ask Question](#)

I wrote a code that basically selects all `input type=text` element like this:

89

```
$('.sys input[type=text]').each(function () {})
```



How do I change it to select `input[type=text]` or `select` ?

17

[javascript](#)[jquery](#)

asked May 18 '12 at 9:15

[Jack](#)

3,170

14

49

99

4 Answers



Using a normal css selector:

158

```
$('.sys input[type=text], .sys select').each(function() {.
```



If you don't like the repetition:



```
$('.sys').find('input[type=text],select').each(function()
```

Or more concisely, pass in the `context` argument:

[Home](#)[PUBLIC](#)[Stack Overflow](#)[Tags](#)[Users](#)[Jobs](#)[Teams](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Note: Internally jQuery will convert the above to `find()` equivalent

<http://api.jquery.com/jQuery/>

Internally, selector context is implemented with the `.find()` method, so `$('span', this)` is equivalent to `$(this).find('span')`.

I personally find the first alternative to be the most readable :), your take though

edited May 18 '12 at 10:02

answered May 18 '12 at 9:16




Andreas Wong

48.2k 16 92 117

-
- 1 Since the context form is using the find form, the find form is more efficient than the context form (one call function avoided). This is valid for almost all selector used. Then, IMO the find form is more efficient than the normal CSS selector, because both parts of the selector are relative to the root node, where in the find form, only the `.sys` part is relative to it, then `input[type=text],select` is executed on a much smaller set of elements so it may be faster (but need this need to be verified by tests) – [pomeh](#) May 20 '12 at 16:55
-
- 1 @pomeh I can see where you are coming from, but if performance of a `$` call is that *important* to your app, please avoid using jQuery altogether :). This answer tried to answer OP's question, if it had been a question of performance, this answer won't be here in the first place. Thanks anyhow for the comment :), appreciate it – [Andreas Wong](#) May 20 '12 at 17:01
-

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

which performs better bc. slow performance === unhappy users . Also, we can **both** answer an OP's question with multiple answers (as you did) and provided advantages/inconvenient of each of them (as I did in comment). IMO it is important to notice why all answers are different, while providing the same result. Also, we can write vanilla JavaScript code that performs slowly: JavaScript != performance – [pomeh](#) May 20 '12 at 17:16

- 1 [@pomeh](#) My point about performance was, if you *really* care about performance, don't use jQuery's `$` , classify your divs specifically and use `document.getElementById/ElementsByClassName` , rather than going through `$` which does a lot of checking / string parsing of your selector, jQuery isn't famous for its performant library. And I haven't honestly seen an app slowing down because of calling `$` one too many, if you have a website that has that problem, please show me, I'm very interested :) – [Andreas Wong](#) May 21 '12 at 0:09 
- 2 [@pomeh](#) Yes, I whole-heartedly agree with your point of JS != Performance :), at the end of the day, it still boils down to us, the programmers to actually write sensible code. Thanks for the short discussion, have a good day :) – [Andreas Wong](#) May 22 '12 at 2:41



```
$('.sys').children('input[type=text], select').each(function
```

6



EDIT: Actually this code above is equivalent to the children selector `.sys > input[type=text]` if you want the descendant select (`.sys input[type=text]`) you need to use the options given by [@NiftyDude](#).

More information:

- [jQuery.children\(\)](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

edited May 18 '12 at 9:54

answered May 18 '12 at 9:16



Wouter J

36.2k 11 89 102

you typed `children` instead of `children` – [pomeh](#) May 18 '12 at 9:17

and I've fixed it – [Andreas Wong](#) May 18 '12 at 9:17

@NiftyDude thank you. I've added some reference links. – [Wouter J](#) May 18 '12 at 9:19



If you have multiple inputs as text in a form or a table that you need to iterate through, I did this:

4



```
var $list = $("#tableOrForm :input[type='text']");

$list.each(function(){
    // Go on with your code.
});
```

What I did was I checked each input to see if the type is set to "text", then it'll grab that element and store it in the jQuery list. Then, it would iterate through that list. You can set a temp variable for the current iteration like this:

```
var $currentItem = $(this);
```

This will set the current item to the current iteration of your `for each` loop. Then you can do whatever you want with the

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Hope this helps anyone!

answered Nov 19 '13 at 14:54



[Jason Cidras](#)

436 3 11



```
$('#input[type=text],select', '.sys');
```

3

for looping:



```
$('#input[type=text],select', '.sys').each(function() {  
    // code  
});
```

answered May 18 '12 at 9:17



[thecodeparadox](#)

71.9k 18 123 156

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).