

How do I disable form fields using CSS?

Asked 8 years, 4 months ago Active 1 year, 7 months ago Viewed 391k times



Is it possible to disable form fields using CSS? I of course know about the attribute disabled, but is it possible to specify this in a CSS rule? Something like -

144



13

```
<input type="text" name="username" value="admin" >
<style type="text/css">
  input[name=username] {
    disabled: true; /* Does not work */
  }
</style>
```

The reason I'm asking is that, I have an application where the form fields are autogenerated, and fields are hidden/shown based on some rules (which run in Javascript). Now I want to extend it to support disabling/enabling fields, but the way the rules are written to directly manipulate the style properties of the form fields. So now I have to extend the rule engine to change attributes as well as style of form fields and somehow it seems less than ideal.

It's very curious that you have visible and display properties in CSS but not enable/disable. Is there anything like it in the still-under-works HTML5 standard, or even something non-standard (browser specific)?

html

css

asked May 11 '11 at 10:57



[Anupam Jain](#)

4,201 2 30 65

4 It's mentioned several times below but it's lost in the noise. Try pointer-events: none; – [Corey Alix](#) Apr 8 '16 at 2:31

12 Answers

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

143

`pointer-events:none;`

You might also want to change colors etc.

answered Mar 10 '14 at 14:35

[ANaimi](#)**3,521**

6

28

29

57 This helps, but doesn't prevent tabbing into the fields. – [jerwood](#) Aug 8 '14 at 19:09

2 not working in IE – [Kuncevič](#) Apr 4 '15 at 2:00

2 This also does not prevent form values from being submitted. – [Ken Wayne VanderLinde](#) Feb 10 '17 at 7:47

1 @KenWayneVanderLinde neither does the disabled attribute right? – [theonlygusti](#) Jan 18 '18 at 12:47

1 @theonlygusti Disabled <input> elements in a form will not be submitted. – [doug_d_in_nc](#) Mar 13 '18 at 4:03



Since the rules are running in JavaScript, why not disable them using javascript (or in my examples case, jQuery)?

103

```
$('#fieldId').attr('disabled', 'disabled'); //Disable
$('#fieldId').removeAttr('disabled'); //Enable
```



UPDATE

The `attr` function is no longer the primary approach to this, as was pointed out in the comments below. This is now done with the `prop` function.

```
$( "input" ).prop( "disabled", true ); //Disable
$( "input" ).prop( "disabled", false ); //Enable
```

edited Aug 13 '15 at 12:27

answered May 11 '11 at 11:00


[Dutchie432](#)**24.2k**

19

85

106

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

- 5 It is probably worth noting that this solution won't work if the user has disabled JavaScript. – [josh-fuggie](#) Apr 18 '13 at 0:02
- 1 Firefox 20.0 (not sure about other versions, this is just what I'm using to develop) requires removing the attribute. While [@Dutchie432](#) is right in theory, using `.removeAttr('disabled');` is the correct practical solution (at least in my current browser). – [user636044](#) Apr 22 '13 at 6:16
- 1 Unless you can't use JQuery. – [marines](#) Apr 24 '14 at 11:30
- 3 FYI - the jQuery documentation suggests using `.prop()` instead of `.attr()` for this case. "As of jQuery 1.6, the `.attr()` method returns undefined for attributes that have not been set. To retrieve and change DOM properties such as the checked, selected, or disabled state of form elements, use the `.prop()` method." Source Documentation: [.attr\(\)](#) and [.prop\(\)](#) – [Nicholas Ryan Bowers](#) May 11 '15 at 19:58 

▲ This can be helpful:

70

```
<input type="text" name="username" value="admin" >
```

```
<style type="text/css">
input[name=username] {
    pointer-events: none;
}
</style>
```

Update:

and if want to disable from tab index you can use it this way:

```
<input type="text" name="username" value="admin" tabindex="-1" >
```

```
<style type="text/css">
input[name=username] {
    pointer-events: none;
}
</style>
```

edited Sep 21 '17 at 19:22



[John Culviner](#)

16k 5 40 44

answered Jun 10 '15 at 10:09



[Fereydoon Barikzehy](#)

1,890 22 32

4 That looked like just what I was after to disable all my input fields but although it blocks them from mouse events, they can still be accessed and

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

11 `tab-index` css property does not exist. It needs to be a html attribute (`tabindex=-1`) – [N4ppeL](#) Dec 2 '15 at 14:05

`tab-index` not find and not work in chrome, hitting TAB key passed away disable feature so this is not complete solution. – [QMaster](#) Aug 17 '16 at 12:38

▲ It's very curious that you have `visible` and `display` properties in CSS but not `enable/disable`.

53



It's very curious that you think that's very curious. You're misunderstanding the purpose of CSS. CSS is not meant to change the **behavior** of form elements. It's meant to change their **style** only. Hiding a text field doesn't mean the text field is no longer there or that the browser won't send its data when you submit the form. All it does is hide it from the user's eyes.

To actually disable your fields, you **must** use the `disabled` attribute in HTML or the `disabled` DOM property in JavaScript.

edited May 11 '11 at 11:07

answered May 11 '11 at 11:01



BoltClock ♦

549k 133 1202
1228

31 Of course I know what CSS is "supposedly" for. But for modern webapps the distinction between "presentation" and "behaviour" is muddier than mud. Since disabled fields are not submitted to the server, it's a behaviour change. But then how is hiding all the form fields not one?! Perhaps in the good old days when all you could do on the web is read text and fill up forms, the distinction was clearer. In the modern webapp, if you want to separate presentation from behaviour, CSS versus HTML/JS is the wrong way to do it. – [Anupam Jain](#) May 11 '11 at 11:58

16 `disabled` is a state, not a behavior. What he's asking for is perfectly reasonable. For instance, you might want to add a busy class to multiple fields while data is being processed. Furthermore, the W3C seems to agree since it allows CSS selection of elements via the `disabled` pseudo class. – [IAmNaN](#) Aug 29 '13 at 20:58

3 @IAmNaN: 1) "disabled is a state, not a behavior." So is just about every other HTML/DOM attribute. 2) "For instance, you might want to add a busy class to multiple fields while data is being processed." You do that in a script. 3) "Furthermore, the W3C seems to agree since it allows CSS selection of elements via the `disabled` pseudo class." Being able to *select* an element based on whether it's enabled or disabled has nothing to do with being able to *change* that state using CSS. – [BoltClock](#) ♦ Aug 30 '13 at 7:30 ✎

4 There are many things in CSS that actually could be considered to be changes in *behavior* e.g. `pointer-events:none`; mentioned by @ANaimi. So `disabled` could be easily considered to be `display` property as well. Too bad it is not, would make few things easier. – [Mikael Lepistö](#) Sep 3 '14 at 7:43

1 @Mikael Lepistö: Yeah, I don't agree with `pointer-events` being a CSS property either. FYI, `pointer-events` [originates from SVG](#). – [BoltClock](#) ♦ Sep 3 '14 at 8:58

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

19

`disabled="disabled"`

answered May 11 '11 at 11:02

[mr_eclair](#)

22k

32

107

139

While this does work in principle, if you read the full question it says the form field are being auto generated, and then need to be disabled afterwards. – [thelem](#) Sep 22 '14 at 15:07

1 @mr_eclair. it is possible using ---> pointer-events: none; Css property. – [Fereydoon Barikzehy](#) Jun 10 '15 at 10:17

3 Note: Elements with `disabled` attribute are **not submitted** - their values are not posted to the server. Read on: stackoverflow.com/q/8925716/1066234 – [Kai Noack](#) Dec 10 '15 at 15:45

The practical solution is to use CSS to actually hide the input.

17

To take this to its natural conclusion, you can write two html inputs for each actual input (one enabled, and one disabled) and then use javascript to control the CSS to show and hide them.

answered May 11 '11 at 11:06

[graphicdivine](#)

9,112

7

27

56

3 If your using Javascript it would make sense to just disable/enable as needed no? – [MindVox](#) Feb 12 '16 at 13:16

Indeed, but in the context of this particular question, this might be a viable approach. – [graphicdivine](#) Feb 12 '16 at 15:01

first time answering something, and seemingly just a bit late...

10

I agree to do it by javascript, if you're already using it.

For a composite structure, like I usually use, I've made a css pseudo after element to block the elements from user interaction, and allow styling without having to manipulate the entire structure.

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

```

<div id=test class=stdInput>
  <label class=stdInputLabel for=selecterthingy>A label for this input</label>
  <label class=selectWrapper>
    <select id=selecterthingy>
      <option selected disabled>Placeholder</option>
      <option value=1>Option 1</option>
      <option value=2>Option 2</option>
    </select>
  </label>
</div>

```

I can place a `disabled` class on the wrapping `div`

```

.disabled {
  position : relative;
  color    : grey;
}
.disabled:after {
  position : absolute;
  left     : 0;
  top      : 0;
  width    : 100%;
  height   : 100%;
  content  : ' ';
}

```

This would grey text within the `div` and make it unusable to the user.

[My example JSFiddle](#)

answered Nov 24 '13 at 22:03



[Larchy](#)

123 2 9

3 BTW: You still need to handle tabbing which still allows the user to get to the field and change it. – [K Kimble](#) Nov 25 '13 at 18:59

1 so shouldn't the `<select>` have a `class="disabled"`? – [TimoSolo](#) Nov 28 '13 at 9:04

This is brilliant I knew there was something like this. :) thanks – [James Harrington](#) Jan 11 '18 at 5:30

This is really useful. I know using `inverted` to set disabled would be better, but this solves my problem of making it clear to users (in addition to an error

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

I am always using:

```
5 input.disabled {  
    pointer-events:none;  
    color:#AAA;  
    background:#F5F5F5;  
}
```

and then applying the css class to the input field:

```
<input class="disabled" type="text" value="90" name="myinput" id="myinput" />
```

answered Apr 23 '15 at 8:08



[Kai Noack](#)

7,262 6 71 126

You cannot do that I'm afraid, but you can do the following in jQuery, if you don't want to add the attributes to the fields. Just place this inside your `<head></head>` tag

```
2 $(document).ready(function(){  
    $(".inputClass").focus(function(){  
        $(this).blur();  
    });  
});
```

If you are generating the fields in the DOM (with JS), you should do this instead:

```
$(document).ready(function(){  
    $(document).on("focus", ".inputClass", function(){  
        $(this).blur();  
    });  
});
```

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



2



```
input[name=username] { disabled: true; /* Does not work */ }
```

I know this question is quite old but for other users who come across this problem, I suppose the easiest way to disable input is simply by ':disabled'

```
<input type="text" name="username" value="admin" disabled />
<style type="text/css">
  input[name=username]:disabled {
    opacity: 0.5 !important; /* Fade effect */
    cursor: not-allowed; /* Cursor change to disabled state*/
  }
</style>
```

In reality, if you have some script to disable the input dynamically/automatically with javascript or jquery that would automatically disable based on the condition you add.

In jQuery for Example:

```
if (condition) {
  // Make this input prop disabled state
  $('input').prop('disabled', true);
}
else {
  // Do something else
}
```

Hope the answer in CSS helps.

answered Jan 18 '18 at 12:45



[Shaze](#)

157 1 9



This can be done for a non-critical purpose by putting an overlay on top of your input element. Here's my example in pure HTML and

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

<https://jsfiddle.net/1tL40L99/>

```
<div id="container">
  <input name="name" type="text" value="Text input here" />
  <span id="overlay"></span>
</div>

<style>
  #container {
    width: 300px;
    height: 50px;
    position: relative;
  }
  #container input[type="text"] {
    position: relative;
    top: 15px;
    z-index: 1;
    width: 200px;
    display: block;
    margin: 0 auto;
  }
  #container #overlay {
    width: 300px;
    height: 50px;
    position: absolute;
    top: 0px;
    left: 0px;
    z-index: 2;
    background: rgba(255,0,0, .5);
  }
</style>
```

answered Feb 25 '15 at 0:14



DeFeNdog

652 1 6 19

There's no way to use CSS for this purpose. My advice is to include a javascript code where you assign or change the css class applied to the inputs. Something like that :

1

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

```
.toggleClass('class_enabled');

$('.class_disabled').attr('disabled', '');
$('.class_enabled').removeAttr('disabled', '');
}

.class_disabled { background-color : #FF0000; }
.class_enabled { background-color : #00FF00; }

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>
<form>
  Input: <input id="id_input1" class="class_enabled" />
  <input type="button" value="Toggle" onclick="change_input()"/>
</form>
```

[Run code snippet](#)[Expand snippet](#)

answered Apr 27 '15 at 14:00



Joel Barba

77 4

protected by Ry- ♦ Sep 7 '17 at 3:57

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?