

How to disable all div content

[Ask Question](#)

I was under the assumption that if I disabled a div, all content got disabled too.

240



However, the content is grayed but I can still interact with it.



Is there a way to do that? (disable a div and get all content disabled also)

62

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

edited Mar 12 '09 at 19:58



IAdapter

32.4k 62 157 230

asked Mar 12 '09 at 18:07



juan

39.3k 46 144 178

24 Answers



Many of the above answers only work on form elements. A simple way to disable any DIV including its contents is to just disable mouse interaction. For example:

414



```
$("#mydiv").addClass("disabledbutton");
```

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

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

Teams

Q&A for work


[Learn More](#)

```
.disabledbutton {
  pointer-events: none;
  opacity: 0.4;
}
```

answered Aug 15 '14 at 14:23

**Kokodoko****9,269** 14 54 97

-
- 10 +1 for correct answer - You just saved me hours of work!!!
tears in eyes and might be in love - It's also supported by all
browsers: caniuse.com/#feat=pointer-events – [tfmontague](#)
Jun 13 '15 at 16:09
-
- 10 I know it's quite late, but still, it is not supported by IE 8, IE 9
and IE 10. Just to let everybody know.
caniuse.com/#feat=pointer-events – [Razort4x](#) Aug 3 '15 at
11:54
-
- 10 This will only disallow mouse events, but the control is still
enabled. – [Mario Levrero](#) May 18 '16 at 9:54
-
- 32 Note: With this solution, you can't interact with this element,
or any children of this element, with a mouse, or on a touch
device. But you can still tab to it with your keyboard. – [Adam](#)
Jul 3 '16 at 9:36
-
- 6 Still accessible through keyboard though. – [Tomer W](#) May 6
'17 at 20:43
-



Use a framework like JQuery to do things like:

145

```
function toggleStatus() {
  if ($('#toggleElement').is(':checked')) {
    $('#idOfTheDIV :input').attr('disabled', true);
  } else {
```

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

[Disable And Enable Input Elements In A Div Block Using jQuery](#) should help you!

As of jQuery 1.6, you should use `.prop` instead of `.attr` for disabling.

edited Jan 11 '12 at 20:11



BNL

6,752 4 20 31

answered Mar 12 '09 at 18:09



Martin K.

3,619 7 31 46

-
- 2 "manually" selecting all inputs... I'll try that, but shouldn't it be sufficient to mark the div as disabled? – [juan](#) Mar 12 '09 at 18:13

When I toggle to un-disable, some pagination buttons that need to remain disabled are also toggled... – [juan](#) Mar 12 '09 at 18:18

You can filter this buttons and do a `".attr('disabled', true);"` every time at them! Just do a `$('#idOfPagination').attr('disabled', true);` after the `if{}else{} construct`. – [Martin K.](#) Mar 12 '09 at 18:21

actually their status is controlled elsewhere, it depends on which page of the list I'm paginating I'm on (they don't always need to be disabled). I'd need somehow of doing it without altering the content control's original status – [juan](#) Mar 12 '09 at 18:23

You can check their status also with jquery and save it. Do: `$('#idOfPagination').is(':disabled') ? disablePagination = false : disablePagination = true;` once in a global area, directly after the page has been loaded. – [Martin K.](#) Mar 12 '09 at 19:23

50

I just wanted to mention [this extension method for enabling and disabling elements](#). I think it's a much cleaner way than adding and removing attributes directly.

Then you simply do:

```
$("#div *").disable();
```

edited May 23 '17 at 12:34



Community ♦

1 1

answered Mar 12 '09 at 21:39



cletus

514k 141 844 915

Yes, I implemented like that – [juan](#) Mar 12 '09 at 21:50

This solution may cause side effects in big pages! (No static reference to a div container / Every underlying element is addressed) – [Martin K.](#) Mar 13 '09 at 8:40

If you are using asp.net you will get a <div disabled="disabled"> when you disable a Panel control. This works for child elements (ie. they become disabled) in IE but not other browsers. You can disable all child form elements in Chrome/Firefox by combining the jquery disable function with...`$("#div[disabled='disabled']:input").disable();` – [Stuart](#) Nov 21 '12 at 12:14 ✎

20

Here is a quick comment for people who don't need a div but just a bloquelement. In HTML5 `<fieldset disabled="disabled"></fieldset>` got the disabled attribute. Every form element in a disabled fieldset is disabled.

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



Michael Hächler

331 2 5

-
- 1 This is a great answer - it allows dynamic items to be spawned in a disabled state so long as they're within the block element rather than testing the disabled state on creation - and elements are truly disabled. – [salmonmoose](#) Jan 12 '17 at 8:42



This is the best answer. It's the most semantically correct, telling the browser that all inputs within this fieldset should be disabled. It honors the keyboard and doesn't need mouse handling JS unregistration. One note, though, as of the time of this comment, Edge won't inherit the disabled attribute value from parent fieldsets inside of another fieldset. –

[Stephen Watkins](#) Nov 17 '18 at 21:46



The disabled attribute is not part of the W3C spec for [DIV elements](#), only for [form elements](#).

16



The jQuery approach suggested by Martin is the only foolproof way you're going to accomplish this.

answered Mar 12 '09 at 18:30



Chris Van Opstal

28.4k 7 62 87

-
- 1 This answer should be the accepted answer, as if you're literally want to really disable DIV. – [Eriawan Kusumawardhono](#) Mar 12 '09 at 21:57

Check @Kokodoko 's Answer. That is the correct one – [Oliver](#) May 19 '15 at 5:35



similar to cletu's solution, but i got an error using that

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



```

$('div *').prop('disabled', true);
// or
$('#the_div_id *').prop('disabled', true);

```

works fine on me

answered Aug 23 '13 at 2:37



[Kokizzu](#)

11.4k 16 89 160



You can use this simple CSS statement to disable events

11

```

#my-div {
  pointer-events: none;
}

```



edited Jun 18 '15 at 18:30

answered Apr 6 '15 at 17:03



[Marcio Mazzucato](#)

5,407 3 49 66



Browsers tested: IE 9, Chrome, Firefox and jquery-1.7.1.min.js

6



```

$(document).ready(function () {
  $('#chkDisableEnableElements').change(function () {
    if ($('#chkDisableEnableElements').is(':checked')) {

```

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

```

        disableElements($('#divDifferentElements')
    });
});

function disableElements(el) {
    for (var i = 0; i < el.length; i++) {
        el[i].disabled = true;

        disableElements(el[i].children);
    }
}

function enableElements(el) {
    for (var i = 0; i < el.length; i++) {
        el[i].disabled = false;

        enableElements(el[i].children);
    }
}

```

answered Feb 20 '13 at 5:47



Syed Nasir Abbas

1,239 12 7

4 ▲ HTML input controls can be disabled using 'disabled' attribute as you know. Once 'disabled' attribute for an input control is set, event handlers associated with such control are not invoked.

▼ You have to simulate above behavior for HTML elements that don't support 'disabled' attribute like div, if you wish.

If you have a div, and you want to support click or a key event on that div, then you have to do two things: 1) When you want to disable the div, set its disabled attribute as usual (just to comply with the convention) 2) In your div's

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

(e.g. just return immediately). If disabled attribute is not set, then do your div's click and/or key event logic.

Above steps are browser independent as well.

answered Nov 30 '12 at 16:29



Sabinia

41 1



One way to achieve this is by adding the disabled prop to all children of the div. You can achieve this very easily:

4



```
$("#myDiv").find("*").prop('disabled', true);
```

`$("#myDiv")` finds the div, `.find("*")` gets you all child nodes in all levels and `.prop('disabled', true)` disables each one.

This way all content is disabled and you can't click them, tab to them, scroll them, etc. Also, you don't need to add any css classes.

answered Jan 17 '17 at 14:13



Blueriver

1,019 2 9 27



I thought I'd chip in a couple of notes.

2



1. `<div>` can be disabled in IE8/9. I assume this is "incorrect", and it threw me off

2. Don't use `.removeProp()`, as it has a permanent effect

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

3. `$("#myDiv").filter("input,textarea,select,button").prop("disabled", true)` is more explicit and will catch some form elements you would miss with `:input`

edited Apr 13 '12 at 19:08

answered Apr 13 '12 at 18:51



Steve11235

2,369 1 13 17



This is for the searchers,

2

The best I did is,



```
$('#myDiv *').attr("disabled", true);  
$('#myDiv *').fadeTo('slow', .6);
```

edited Jun 23 '16 at 7:14

answered Jun 23 '16 at 5:24



Shree Krishna

6,556 4 28 60



As mentioned in comments, you are still able to access element by navigating between elements by using tab key. so I recommend this :

2



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

answered Sep 9 '17 at 15:55



Hossein Khalesi

61 7



I would use an improved version of Cletus' function:

1



```
$.fn.disable = function() {
  return this.each(function() {
    if (typeof this.disabled != "undefined") {
      $(this).data('jquery.disabled', this.disabled);

      this.disabled = true;
    }
  });
};

$.fn.enable = function() {
  return this.each(function() {
    if (typeof this.disabled != "undefined") {
      this.disabled = $(this).data('jquery.disabled');
    }
  });
};
```

Which stores the original 'disabled' property of the element.

```
$('#myDiv *').disable();
```

answered Aug 27 '12 at 14:44



koosvdkolk

316 2 4

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



How to disable the contents of a DIV

The CSS `pointer-events` property alone doesn't disable child elements from scrolling, and it's not supported by IE10 and under for DIV elements (only for SVG).

<http://caniuse.com/#feat=pointer-events>

To disable the contents of a DIV on all browsers.

Javascript:

```
$("#myDiv")
  .addClass("disable")
  .click(function () {
    return false;
  });
```

Css:

```
.disable {
  opacity: 0.4;
}
// Disable scrolling on child elements
.disable div,
.disable textarea {
  overflow: hidden;
}
```

To disable the contents of a DIV on all browsers, except IE10 and under.

Javascript:

```
$("#myDiv").addClass("disable");
```

Css:

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

```
opacity: 0.4;
}
// Disable scrolling on child elements
.disable div,
.disable textarea {
  overflow: hidden;
}
```

edited Jun 13 '15 at 19:06

answered Jun 13 '15 at 19:01



[tfmontague](#)

8,138 1 34 31



Below is a more comprehensive solution to masking divs enabling

1



- no separate CSS
- cover the whole page or just an element
- specify mask color and opacity
- specify Z-index so you can show popups over the mask
- show an hourglass cursor over the mask
- removing the masking div on maksOff so a different one can be shown later
- stretch mask when element resize
- return the mask element so you can style it etc

Also included is hourglassOn and hourglassOff which can be used separately

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

```

// elemOrId - jquery element or element id, defaults to $(
// settings.color defaults to 'transparent'
// settings.opacity defaults to 1
// settings.zIndex defaults to 2147483647
// if settings.hourglass==true change cursor to hourglass
function maskOn(elemOrId, settings) {
    var elem=elemFromParam(elemOrId);
    if (!elem) return;

    var maskDiv=elem.data('maskDiv');
    if (!maskDiv) {
        maskDiv=$('<div style="position:fixed;display:inline-block;
        $("body").append(maskDiv);
        elem.data('maskDiv', maskDiv);
    }

    if (typeof settings==='undefined' || settings===null) :
    if (typeof settings.color==='undefined' || settings.co
settings.color='transparent';
    if (typeof settings.opacity==='undefined' || settings.o
settings.opacity=1;
    if (typeof settings.zIndex==='undefined' || settings.z
settings.zIndex=2147483647;
    if (typeof settings.hourglass==='undefined' || setting
settings.hourglass=false;

    // stretch maskdiv over elem
    var offsetParent = elem.offsetParent();
    var widthPercents=elem.outerWidth()*100/offsetParent.o
    var heightPercents=elem.outerHeight()*100/offsetParent
    maskDiv.width(widthPercents);
    maskDiv.height(heightPercents);
    maskDiv.offset($(elem).offset());

    // set styles
    maskDiv[0].style.backgroundColor = settings.color;
    maskDiv[0].style.opacity = settings.opacity;
    maskDiv[0].style.zIndex = settings.zIndex;

    if (settings.hourglass) hourglassOn(maskDiv);

    return maskDiv;
}

// elemOrId - jquery element or element id, defaults to $(

```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```

    var maskDiv=elem.data('maskDiv');
    if (!maskDiv) {
        console.log('maskOff no mask !');
        return;
    }

    elem.removeData('maskDiv');
    maskDiv.remove();
}

// elemOrId - jquery element or element id, defaults to $(
// if decendents is true also shows hourglass over decen
true
function hourglassOn(elemOrId, decendents) {
    var elem=elemFromParam(elemOrId);
    if (!elem) return;

    if (typeof decendents==='undefined' || decendents===nu:

        if ($('#style:contains("hourGlass")').length < 1) $('<s
        cursor: wait !important; }').appendTo('head');
        if ($('#style:contains("hourGlassWithDecendents")').len
        $('<style>').text('.hourGlassWithDecendents, .hourGlassWitl
        !important; }').appendTo('head');
        elem.addClass(decendents ? 'hourGlassWithDecendents' :
    }

// elemOrId - jquery element or element id, defaults to $(
function hourglassOff(elemOrId) {
    var elem=elemFromParam(elemOrId);
    if (!elem) return;

    elem.removeClass('hourGlass');
    elem.removeClass('hourGlassWithDecendents');
}

function elemFromParam(elemOrId) {
    var elem;
    if (typeof elemOrId==='undefined' || elemOrId===null)
        elem=$('#body');
    else if (typeof elemOrId === 'string' || elemOrId inst:
        elem=$('#'+elemOrId);
    else
        elem=$(elemOrId);
}

```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```

    }

    return elem;
}

```

With this you can do for example:

```

maskOn(); // transparent page mask
maskOn(null, {color:'gray', opacity:0.8}); // gray page ma
maskOff(); // remove page mask
maskOn(div); // transparent div mask
maskOn(divId, {color:'gray', hourglass:true}); // gray div
maskOff(div); // remove div mask

```

see [jsfiddle](#)

edited Nov 30 '15 at 0:27

answered Nov 20 '15 at 0:58



kofifus

3,853 3 34 62

Your solutions is very well to disable whole page but it's not working on particular div portion dear, I have tried. – [3 rules](#)
Sep 14 '16 at 11:37



If you wanted to keep the semantics of disabled as follows

1

```
<div disabled="disabled"> Your content here </div>
```



you could add the following CSS

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

```
opacity: 0.4;
}
```

the benefit here is that you're not working with classes on the div that you want to work with

answered Oct 24 '17 at 15:03



[real_ate](#)

5,943 3 21 40



```
function disableItems(divSelector){
    var disableInputs = $(divSelector).find(":input").not(
        disableInputs.attr("data-reenable", true);
        disableInputs.attr("disabled", true);
    )
}

function reEnableItems(divSelector){
    var reenableInputs = $(divSelector).find("[data-reenab:
    reenableInputs.removeAttr("disabled");
    reenableInputs.removeAttr("data-reenable");
}
```

answered Mar 8 '18 at 16:40



[Jtbs](#)

141 8



Another way, in jQuery, would be to get the inner height, inner width and positioning of the containing DIV, and simply overlay another DIV, transparent, over the top the same size. This will work on all elements inside that container, instead of only the inputs.

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

above answers too.

answered Jan 15 '12 at 6:44



[TheCarver](#)

10.9k 20 78 134

What if the user tabs through the controls? This doesn't help at all unless you **ONLY** have users that use the mouse to navigate. – [Sivvy](#) Jan 25 '12 at 14:38

But it can be useful in conjunction with disabling the inputs. If the overlaying div is styled as translucent, it is a good visual indicator that the section is disabled. – [xr280xr](#) Jul 2 '15 at 23:15



`$("#yourdivid textarea, #yourdivid input, #yourdivid select"`

0



edited Sep 20 '13 at 6:20



[Soner Gönül](#)

81.5k 27 150 286

answered Sep 20 '13 at 5:31



[anand](#)

452 5 9



This css only/noscript solution adds an overlay above a fieldset (or a div or any other element), preventing interaction:

0



```
fieldset { position: relative; }
fieldset[disabled]::after { content: ''; display: inline-b.
```

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

If you want an invisible i.e. transparent overlay, set the background to e.g. `rgba(128,128,128,0)`, as it won't work without a background. The above works for IE9+. The following much simpler css will work on IE11+

```
[disabled] { pointer-events: none; }
```

Chrome

edited Sep 9 '14 at 13:00

answered Sep 9 '14 at 12:41



Costas

106 4



0



If you are simply trying to stop people clicking and are not horrifically worried about security - I have found an absolute placed div with a z-index of 99999 sorts it fine. You can't click or access any of the content because the div is placed over it. Might be a bit simpler and is a CSS only solution until you need to remove it.

answered Apr 19 '16 at 11:57



Ukuser32

1,146 2 12 24



the simplest solution

0

Look at my solution

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

the `fieldsetUserInfo` is div contains all inputs I want to disabled or Enable

hope this helps you

answered Nov 24 '16 at 9:55



Basheer AL-MOMANI

7,136 1 50 67



EDIT: Below I've used `.on()` method, instead use

`.bind()` method

-1



```
$(this).bind('click', false);
$(this).bind('contextmenu', false);
```

to remove your setting, you can use `.unbind()` method. Whereas the `.off()` method doesn't work as expected.

```
$(this).unbind('click', false);
$(this).unbind('contextmenu', false);
```

After researching hundreds of solutions! learning about pointer-events, below is what I did.

As [@Kokodoko](#) mentioned in his solution which is apt for all browsers except IE. `pointer-events` work in **IE11** and not in the lower versions. I also noticed in **IE11**, pointer-events do not work on the child elements. And hence if we have something like below

```
<a href="www.preshmalinetpereira.wordpress.com"><i class=
```

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

where span *-is the child element*, setting `pointer-events: none` wont work

To overcome this problem I wrote a function which could act as pointer-events for IE and will work in the lower versions.

In JS File

```
DisablePointerEvents(".DisablePointerEvents");

function DisablePointerEvents(classId) {
    $(classId).each(function () {
        $(this).on('click', false );
        $(this).on('contextmenu', false );
    });
}
```

In CSS File

```
.DisablePointerEvents{
    pointer-events: none;
    opacity: 0.7;
    cursor: default;
}
```

In HTML

```
<a href="www.preshmalinetpereira.wordpress.com" class="Di:
class="car icon"></i><span>My Blog</span></a>
```

This faked the `pointer-events` scenario where `pointer-events` doesnt work and when the above condition of child elements occur.

JS Fiddle for the same

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

edited May 23 '17 at 11:55



Community ♦

1 1

answered Jun 28 '16 at 10:44



Linet Pereira

120 7

protected by [Engineer](#) Sep 9 '14 at 12:56

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?