What's the difference between disabled="disabled" and readonly="readonly" for HTML form input fields?

Asked 7 years, 11 months ago   Active 7 months ago   Viewed 204k times

▲

390

▼

I have read a bit on this, but I can't seem to find anything solid about how different browsers treat things.

html   cross-browser

⭐

121

edited Dec 11 '18 at 14:42   asked Oct 11 '11 at 18:28

Mark Amery   Andy
72.5k   36   279   324   **3,968**   5   22   50

Andy: You mention textarea, but can't we generalize your question to all HTML form input fields? The (v good) answer from @oezi seems to do so. I'll update your question if that's ok with you. – Adrien Be Jan 24 '14 at 14:03

related: "how to emulate the readonly attribute for a select tag, and still get the POST data?" stackoverflow.com/questions/368813/… – Adrien Be Jan 24 '14 at 14:15

5 Answers

▲

667

▼

A `readonly` element is just not editable, but gets sent when the according `form` submits. A `disabled` element isn't editable and isn't sent on submit. Another difference is that `readonly` elements can be focused (and getting focused when "tabbing" through a form) while `disabled` elements can't.

Read more about this in ~~this great article~~ or the definition by w3c. To quote the important part:

✓   **Key Differences**

The Disabled attribute

- Values for disabled form elements are not passed to the processor method. The W3C calls this a successful element.(This works similar to form check boxes that are not checked.)

- Some browsers may override or provide default styling for disabled form elements. (Gray out or emboss text) Internet Explorer 5.5 is particularly nasty about this.

- Disabled form elements do not receive focus.

- Disabled form elements are skipped in tabbing navigation.

The Read Only Attribute

- Not all form elements have a readonly attribute. Most notable, the `<SELECT>` , `<OPTION>` , and `<BUTTON>` elements do not have readonly attributes (although they both have disabled attributes)

- Browsers provide no default overridden visual feedback that the form element is read only. (This can be a problem… see below.)

- Form elements with the readonly attribute set will get passed to the form processor.

- Read only form elements can receive the focus

- Read only form elements are included in tabbed navigation.

<table>
<tr><td>edited Feb 6 at 16:04</td><td>answered Oct 11 '11 at 18:30</td></tr>
<tr><td>yuriy636
5,590   4   22   34</td><td>oezi
43.9k   9   86   112</td></tr>
</table>

---

5   on readonly element you can't use CTRL + C but you can use right mouse click and select Copy. – Rumplin Jan 22 '14 at 10:49

6   @Rumplin are you sure about that? I just tested and was able to copy with the keyboard shortcut in Chrome on OS X. – evanrmurphy Jul 9 '14 at 15:43

5   "Not all form elements have a readonly attribute. Most notable, the <SELECT> , <OPTION> , and <BUTTON> elements do not have readonly attributes (although thy both have disabled attributes)". Thats why sometimes you have to use "disabled" attribute with a hidden input field for select forms. – Donato May 26 '15 at 18:54

5   So it is my understanding that `disabled` *implies* `readonly` but `readonly` does not imply `disabled` . In other words if an element has the `disabled` attribute then there is no need to also include the `readonly` attribute. Correct? – chharvey Dec 6 '15 at 18:43

1   Wayback archive of that article: web.archive.org/web/20150620041010/https://… – Sam Hasler May 26 '17 at 12:16

88

None of the below will get triggered.

```
$("[disabled]").click( function(){ console.log("clicked") });//No Impact
$("[disabled]").hover( function(){ console.log("hovered") });//No Impact
$("[disabled]").dblclick( function(){ console.log("double clicked") });//No Impact
```

While readonly will be triggered.

```
$("[readonly]").click( function(){ console.log("clicked") });//log - clicked
$("[readonly]").hover( function(){ console.log("hovered") });//log - hovered
$("[readonly]").dblclick( function(){ console.log("double clicked") });//log - double
clicked
```

answered Sep 14 '13 at 4:59

Hrishabh Gupta
1,470   12   9

33

Disabled means that no data from that form element will be submitted when the form is submitted. Read-only means any data from within the element will be submitted, but it cannot be changed by the user.

For example:

```
<input type="text" name="yourname" value="Bob" readonly="readonly" />
```

This will submit the value "Bob" for the element "yourname".

```
<input type="text" name="yourname" value="Bob" disabled="disabled" />
```

This will submit nothing for the element "yourname".

answered Oct 11 '11 at 18:30

Michael Irigoyen
17.9k   15   76   115

5   Both are semantically correct. HTML5 allows you to use either. – Michael Irigoyen Jan 6 '14 at 19:22

11   Then those developers should read the spec. There is nothing wrong with the examples in my answer. – Michael Irigoyen Jan 7 '14 at 2:09

3   Context-free admonition to use only the attribute names, without values, is ill-advised - because then the code will not be valid XML/XHTML. I know a lot of devs don't care about those, but they should at least be aware of the pitfall. Personally I strive for XHTML compliance - unless given a compelling reason, which I haven't received yet - so I use the long/duplicated form. – underscore_d Sep 3 '15 at 12:15 ✎

1   @ToolmakerSteve Do you have a spec citation that empty strings are valid XHTML? I can only find commentary pages saying it's valid for HTML5. Everyone I've seen talking about XHTML say that its form for boolean attributes must be `attrname="attrname"`. Either way, it doesn't seem to be particularly well-documented, at least not that I can find. Well, there's this - w3.org/TR/html4/intro/sgmltut.html#h-3.3.4.2 - but it only specifically mentions SGML and HTML, not XHTML.... too many acronyms :S – underscore_d Sep 1 '16 at 11:14 ✎

9

Elements with Disabled attribute would not submitted or can say their values would not posted with request.

i.e.

```html
<input type="textbox" name="field" value="field" disabled="disabled" />
```

Difference

- Disabled controls do not receive focus.
- Disabled controls are skipped in tabbing navigation.
- Disabled controls cannot be successfully posted.

Use readonly attribute in case you want to post your field's data.

i.e.

```html
<input type="textbox" name="field" value="field" readonly="readonly" />
```

- Read-only elements receive focus but cannot be modified by the user.
- Read-only elements are included in tabbing navigation.
- Read-only elements are successfully posted.

Same as the other answers (disabled isn't sent to the server, readonly is) but some browsers prevent highlighting of a disabled form, while read-only can still be highlighted (and copied).

http://www.w3schools.com/tags/att_input_disabled.asp

http://www.w3schools.com/tags/att_input_readonly.asp

> A read-only field cannot be modified. However, a user can tab to it, highlight it, and copy the text from it.

answered Oct 11 '11 at 18:33

Tim
13.3k   6   34   61

9   *sidenote:* You trust w3schools? oops. Read w3fools.com – Raptor Jan 6 '14 at 12:02

2   @Raptor w3schools is not frowned upon any longer. From w3fools.com: *W3Schools still has issues but they have at least worked on the primary concern developers had. For many beginners, W3Schools has structured tutorials and playgrounds that offer a decent learning experience.* – Daniel Lidström Jan 27 '17 at 10:11