textarea character limit

Asked 13 years, 8 months ago Modified 6 years, 9 months ago Viewed 144k times

▲

37

▼

I would like to be able to limit the number of characters in a textarea. The method I am using works great in Google Chrome, but is slow in Firefox, and doesn't work in IE.

Javascript:

```
function len(){
  t_v=textarea.value;
  if(t_v.length>180){
    long_post_container.innerHTML=long_post;

post_button.className=post_button.className.replace('post_it_regular','post_it_dis
    post_button.disabled=true;
  }
  else{
    long_post_container.innerHTML="";

post_button.className=post_button.className.replace('post_it_disabled','post_it_re
    post_button.disabled=false;
  }
  if(t_v.length>186){
        t_v=t_v.substring(0,186);
    }
}
```

HTML:

```
<textarea id="user_post_textarea" name="user_post_textarea" cols="28" rows="1"
onkeypress="len();" onkeyup="len();"></textarea>
```

Javascript at bottom of body element:

```
textarea=document.getElementById('user_post_textarea');
```

javascript html textarea cross-browser

Share

Improve this question

Follow

edited Oct 18, 2017 at 4:46

asked Apr 3, 2011 at 22:49

Web_Designer

74.4k ● 93 ● 208 ● 266

1 And...? How can this question be answered if we can't even see the code? – Yi Jiang Apr 3, 2011 at 22:49

1 update your question with more details please – user455318 Apr 3, 2011 at 22:50

2 Plus, if you don't want people laughing at you, don't put "html5" and "browser compatibility" in the same sentence.. – Christian Apr 3, 2011 at 22:52

10 Answers

Sorted by: Highest score (default)

92

✓

I found a good solution that uses the maxlength attribute if the browser supports it, and falls back to an unobtrusive javascript pollyfill in unsupporting browsers.

Thanks to @Dan Tello's comment I fixed it up so it works in IE7+ as well:

HTML:

```
<textarea maxlength="50" id="text">This textarea has a character limit of 50.
</textarea>
```

Javascript:

```
function maxLength(el) {
    if (!('maxLength' in el)) {
        var max = el.attributes.maxLength.value;
        el.onkeypress = function () {
            if (this.value.length >= max) return false;
        };
    }
}

maxLength(document.getElementById("text"));
```

Demo

There is no such thing as a `minlength` attribute in HTML5.
For the following input types: `number`, `range`, `date`, `datetime`, `datetime-local`, `month`, `time`, and `week` (which aren't fully supported yet) use the `min` and `max` attributes.

Share
Improve this answer
Follow

edited May 23, 2017 at 12:18
Community Bot
1 ● 1

answered Apr 9, 2011 at 18:36
Web_Designer
74.4k ● 93 ● 208 ● 266

You edited method seems to be busted in IE 7 + 8. First one seems to work though.
– Dan Tello Apr 10, 2012 at 17:45

@DanTello New code with a new demo. Works in IE7+ and other browsers. haven't checked in IE6-. – Web_Designer Dec 2, 2012 at 9:35

Not working in ie "Invalid operand to 'in': Object expected " – Nilesh patel Feb 25, 2013 at 7:17

@Nileshpatel I've tested my demo (Here's a fullscreen version for IE testing) in IE7+ with no errors in the console, and the character limit working properly. – Web_Designer Feb 25, 2013 at 20:28

5 Problem is: if you run into the limit, it doesn't accept *any* key press anymore, so you can't even backspace the last sentence. You must use the context menu to delete something. – DanMan May 29, 2013 at 11:44

8

~~This is entirely untested but it should do what you need.~~

Update : here's a jsfiddle to look at. Seems to be working. link

You would past it into a js file and reference it after your jquery reference. You would then call it like this..

```
$("textarea").characterCounter(200);
```

A brief explanation of what is going on..

On every keyup event the function is checking what type of key is pressed. If it is acceptable the the counter will check the count, trim any excess and prevent any further input once the limit is reached.

The plugin should handle pasting into the target too.

```
  ; (function ($) {
        $.fn.characterCounter = function (limit) {
            return this.filter("textarea, input:text").each(function () {
                var $this = $(this),
                  checkCharacters = function (event) {

                      if ($this.val().length > limit) {

                          // Trim the string as paste would allow you to make it
                          // more than the limit.
                          $this.val($this.val().substring(0, limit))
                          // Cancel the original event
                          event.preventDefault();
                          event.stopPropagation();

                      }
                  };

                $this.keyup(function (event) {
                    // Keys "enumeration"
                    var keys = {
                        BACKSPACE: 8,
                        TAB: 9,
                        LEFT: 37,
                        UP: 38,
                        RIGHT: 39,
                        DOWN: 40
                    };

                    // which normalizes keycode and charcode.
                    switch (event.which) {

                        case keys.UP:
                        case keys.DOWN:
                        case keys.LEFT:
                        case keys.RIGHT:
                        case keys.TAB:
                            break;
                        default:
                            checkCharacters(event);
                            break;
                    }

                });

                // Handle cut/paste.
                $this.bind("paste cut", function (event) {
                    // Delay so that paste value is captured.
                    setTimeout(function () { checkCharacters(event); event = null; }, 150);
                });
            });
        };
    } (jQuery));
```

Share

Improve this answer

Follow

edited Apr 4, 2011 at 0:15

answered Apr 4, 2011 at 0:00

James South

10.6k ● 4 ● 63 ● 118

In your example then when a key is held down the textarea doesn't trim the number of characters until the key is released. I would like something like @webarto has, but that will filter out [DEL] and [BACKSPACE]. – Web_Designer Apr 4, 2011 at 0:33

The example you have allows the placement of another character before it removes this character half a second later but this isn't really what I wanted. I edited the answer @Christian Sciberras provided to filter necessary keycodes and it works great! – Web_Designer Apr 4, 2011 at 4:05

@inquisitive: That could have been fixed easily by changing the line that says if ($this.val().length > limit) to ($this.val().length >= limit). I hope to god you are not writing it in-line like in Christian's example and have at least moved it into a separate js file. – James South Apr 4, 2011 at 11:57

Start using `event.key` instead of `keyCode`, `charCode`, or `which` for standards compliance (the latter ones are all deprecated). – Marcel Waldvogel Oct 3, 2022 at 11:07

2

using maxlength attribute of textarea would do the trick ... simple html code .. not JS or JQuery or Server Side Check Required....

Share Improve this answer Follow

answered Jun 6, 2013 at 8:02

Rohit Gupta
29 ● 2

1 Older browsers don't support maxlength. For IE, it must be version 10. – jplandrain Sep 12, 2014 at 8:25

1

This works on keyup and paste, it colors the text red when you are almost up to the limit, truncates it when you go over and alerts you to edit your text, which you can do.

var t2= /* textarea reference*/

```
t2.onkeyup= t2.onpaste= function(e){
    e= e || window.event;
    var who= e.target || e.srcElement;
    if(who){
        var val= who.value, L= val.length;
        if(L> 175){
            who.style.color= 'red';
        }
        else who.style.color= ''
        if(L> 180){
            who.value= who.value.substring(0, 175);
            alert('Your message is too long, please shorten it to 180
characters or less');
            who.style.color= '';
        }
    }
}
```

Share Improve this answer Follow

answered Apr 4, 2011 at 4:05

kennebec
105k ● 32 ● 108 ● 127

I think that doing this might be easier than most people think!

Try this:

1

```
var yourTextArea = document.getElementById("usertext").value;
// In case you want to limit the number of characters in no less than, say, 10
// or no more than 400.
    if (yourTextArea.length < 10 || yourTextArea.length > 400) {
        alert("The field must have no less than 10 and no more than 400
characters.");
        return false;
    }
```

Please let me know it this was useful. And if so, vote up! Thx!

Daniel

Share Improve this answer Follow

answered Nov 27, 2012 at 1:42

Daniel Montenegro
1,549 ● 2 ● 15 ● 16

1

I believe if you use delegates, it would work..

```
$("textarea").on('change paste keyup', function () {
    var currText = $(this).val();
    if (currText.length > 500) {
        var text = $(this).text();
        $(this).text(text.substr(0, 500));
        alert("You have reached the maximum length for this field");
    }
});
```

Share

Improve this answer

Follow

edited Mar 13, 2018 at 20:22

sorak
2,633 ● 2 ● 18 ● 24

answered Mar 13, 2018 at 20:01

Claude Torrens
21 ● 5

0

Try using jQuery to avoid cross browser compatibility problems...

```
$("textarea").keyup(function(){
    if($(this).text().length > 500){
        var text = $(this).text();
        $(this).text(text.substr(0, 500));
    }
});
```

Share

Improve this answer

Follow

edited Apr 3, 2011 at 23:23

answered Apr 3, 2011 at 22:56

Dejan Marjanović
19.4k ● 7 ● 53 ● 67

Yes, but just to be safe. Anyway he needs to check it on submit and on server side. – Dejan Marjanović Apr 3, 2011 at 23:00

That makes it very slow. Consider that you are move contents on each keypress, even cursor movement. – Christian Apr 3, 2011 at 23:03

all you need to do is .keyup(function(e) { if (>500) { e.preventDefault() }); – MikeMurko Nov 18, 2012 at 17:23

0

Quick and dirty universal jQuery version. Supports copy/paste.

```
$('textarea[maxlength]').on('keypress mouseup', function(){
    return !($(this).val().length >= $(this).attr('maxlength'));
});
```

Share

Improve this answer

Follow

edited Feb 16, 2017 at 17:29

answered Feb 16, 2017 at 17:19

Nick B

507 ●5 ●8

-1

I have written my method for this, have fun testing it on your browsers, and give me feedback.

cheers.

```
function TextareaControl(textarea,charlimit,charCounter){
    if(
            textarea.tagName.toLowerCase() === "textarea" &&
            typeof(charlimit)==="number" &&
            typeof(charCounter) === "object" &&
            charCounter.innerHTML !== null
    ){
        var oldValue = textarea.value;
        textarea.addEventListener("keyup",function(){
            var charsLeft = charlimit-parseInt(textarea.value.length,10);
            if(charsLeft<0){
                textarea.value = oldValue;
                charsLeft = charlimit-parseInt(textarea.value.length,10);
            }else{
                oldValue = textarea.value;
            }
            charCounter.innerHTML = charsLeft;
        },false);
    }
}
```

Share Improve this answer Follow

answered Apr 3, 2011 at 22:52

Jakob Alexander Eichler
3,056 ● 3 ● 34 ● 51

▲

-1

▼

```
... onkeydown="if(value.length>500)value=value.substr(0,500);
if(value.length==500)return false;" ...
```

It ought to work.

Share

Improve this answer

Follow

edited Apr 3, 2011 at 23:01

answered Apr 3, 2011 at 22:56

Christian
28.1k ● 17 ● 116 ● 159

1 It will match exactly 500, what if user pastes text and then continue typing?
– Dejan Marjanović Apr 3, 2011 at 22:58

What does returning false do in this instance? Does it not allow additional text to be typed in the textarea? – Web_Designer Apr 3, 2011 at 23:03

Exactly. Returning false aborts the event. In jQuery there's something like "stopPropagation" or something. – Christian Apr 3, 2011 at 23:05

Ok, So I implimented your example but when I reached the character limit (500) it just stopped and I couldn't even back space or delete part of the text. – Web_Designer Apr 3, 2011 at 23:14

Right, you would want to filter out [DEL] and [BACKSPACE] – Christian Apr 3, 2011 at 23:23