

Applying an ellipsis to multiline text

[Ask Question](#)

Is there a solution to add ellipsis on last line inside a div with a fluid height (20%)?

I found the `-webkit-line-clamp` function in CSS, but in my case the line number will be depending on window size.

```
p {  
  width:100%;  
  height:20%;  
  background:red;  
  position:absolute;  
}
```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed dui felis. Vivamus vitae pharetra nisl, eget fringilla elit. Ut nec est sapien. Aliquam dignissim velit sed nunc imperdiet cursus. Proin arcu diam, tempus ac vehicula a, dictum quis nibh. Maecenas vitae quam ac mi venenatis vulputate. Suspendisse fermentum suscipit eros, ac ultricies leo sagittis quis. Nunc sollicitudin lorem eget eros eleifend facilisis. Quisque bibendum sem at bibendum suscipit. Nam id tellus mi. Mauris vestibulum, eros ac ultrices lacinia, justo est faucibus ipsum, sed sollicitudin sapien odio sed est. In massa ipsum, bibendum quis lorem et, volutpat ultricies nisi. Maecenas scelerisque sodales ipsum a hendrerit Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed dui felis. Vivamus vitae pharetra nisl, eget fringilla elit. Ut nec est sapien. Aliquam dignissim velit sed nunc imperdiet cursus. Proin arcu diam, tempus ac vehicula a, dictum quis nibh. Maecenas vitae quam ac mi venenatis vulputate. Suspendisse fermentum suscipit eros, ac ultricies leo sagittis quis. Nunc sollicitudin lorem eget eros eleifend facilisis. Quisque bibendum sem at bibendum suscipit. Nam id tellus mi. Mauris vestibulum, eros ac ultrices lacinia, justo est faucibus ipsum, sed sollicitudin sapien odio sed est. In massa ipsum, bibendum quis lorem et, volutpat ultricies nisi. Maecenas scelerisque sodales ipsum a hendrerit.</p>

[Run code snippet](#)[Copy snippet to answer](#)[Expand](#)[snippet](#)

I have this JSFiddle to illustrate the issue. <https://jsfiddle.net/96knodm6/>

[html](#)[css](#)[css3](#)[ellipsis](#)

edited Jul 12 '16 at 23:21



Michael_B

141k 45 225 335

asked Oct 10 '15 at 19:18



Bruno Landowski

660 1 7 16

- 3 Possible duplicate of [Limit text length to n lines using CSS](#) – user663031 Oct 24 '15 at 13:50

Did you search for other answers to this question? It's come up a bunch of times. In addition to the question suggested as a duplicate, have a look at stackoverflow.com/questions/29160687/... – user663031 Oct 24 '15 at 13:51

Your solution doesn't work for fluid height as we don't know exactly the number of lines depending on screen sizes. The only solution I found is to add a blurry div at the bottom to middle hide the last line. – Bruno Landowski Oct 26 '15 at 9:31

20 Answers

If you want to apply ellipsis (...) to a *single line of text*, CSS makes that somewhat easy with the `text-overflow` property. It's still a bit tricky (due to all the requirements – see below), but `text-overflow` makes it possible and reliable.

If, however, you want to use ellipsis on *multiline text* – as would be the case here – then don't expect to have any fun. CSS has no standard method for doing this, and the workarounds are hit and miss.

Ellipsis for Single Line Text

With `text-overflow`, ellipsis can be applied to a single line of text. The following CSS requirements must be met:

- must have a `width`, `max-width` or `flex-basis`
- must have `white-space: nowrap`
- must have `overflow` with value other than `visible`
- must be `display: block` or `inline-block` (or the functional equivalent, such as a flex item).

So this will work:

```
p {
  width: 200px;
  white-space: nowrap;
  overflow: hidden;
  display: inline-block;
  text-overflow: ellipsis;
  border: 1px solid #ddd;
  margin: 0;
}

<p>
  This is a test of CSS <i>text-overf]
  This is a test of CSS <i>text-overf]
  This is a test of CSS <i>text-overf]
  This is a test of CSS <i>text-overf]
  This is a test of CSS <i>text-overf]
  This is a test of CSS <i>text-overf]
</p>
```

Run code snippet

Copy snippet to answer

[Expand](#)

[snippet](#)

[jsFiddle version](#)

BUT, try removing the `width`, or letting the `overflow` default to `visible`, or removing `white-space: nowrap`, or using something other than a block container element, AND, ellipsis fails miserably.

One big takeaway here: *text-overflow: ellipsis has no effect on multiline text.* (The `white-space: nowrap` requirement alone eliminates that possibility.)

[Show code snippet](#)

[jsFiddle version](#)

Ellipsis for Multiline Text

Because CSS has no property for ellipsis on multiline text, various workarounds have been created. Several of these methods can be found here:

- [jQuery dotdotdot...](#)
- [Line Clampin' \(Truncating Multiple Line Text\)](#)
- [CSS Ellipsis: How to Manage Multi-Line Ellipsis in Pure CSS](#)

- [A pure CSS solution for multiline text truncation](#)

edited Jan 6 '17 at 19:27

answered Oct 11 '15 at 2:50



Michael_B

141k 45 225 335

-
- 1 Another method by [Natalia Onischuk](#). – [Dan Dascalescu](#) Jun 13 '16 at 2:59
 - 2 7kb for a jQuery plugin when all the user wants is ... added to their text?? Seriously, seems like someone needs to learn to design plugins to be more efficient. – [AndroidDev](#) Jun 10 '17 at 11:07
 - 4 @AndroidDev You need to know where to cut the text. Measuring text and clipping regions is not simple. Try to measure only the last line of a wrapped block of text. You can't do it easily. One technique is to append a 0-width inline element onto the end of the paragraph and get that element's x position. That's still not even precise, because there is space between the last character and the 0-width element. On top of that these plugins support multiple browsers, where the techniques to measure the text and boundaries are different. This isn't as simple as just adding "..." to the text. – [Gavin](#) Jun 20 '17 at 16:44
-

- 1 +1 [CSS Ellipsis: How to Manage Multi-Line Ellipsis in Pure CSS](#) Best cross-browser solution as of this writing. – [gfullam](#) Sep 29 '17 at 20:56

Lodash also has a very nice [truncate](#) method that will add ellipsis to multi-line text. – [user](#) Mar 1 at 12:10

Please check this css for ellipsis to multi-line text

```
body {  
  margin: 0;
```

```
padding: 50px;
}

/* mixin for multiline */
.block-with-text {
  overflow: hidden;
  position: relative;
  line-height: 1.2em;
  max-height: 6em;
  text-align: justify;
  margin-right: -1em;
  padding-right: 1em;
}
.block-with-text:before {
  content: '...';
  position: absolute;
  right: 0;
  bottom: 0;
}
.block-with-text:after {
  content: '';
  position: absolute;
  right: 0;
  width: 1em;
  height: 1em;
  margin-top: 0.2em;
  background: white;
}
```

`<p class="block-with-text">`The Hitch Hiker's Guide to the Galaxy has a few things to say on the subject of towels. A towel, it is said, is the most useful item you can have. Paraphrase it around you for warmth as you bound across it on the brilliant marble-sanded beaches of the planet; use it to mop up hot vapours; you can sleep under it beneath the stars; use it to cover the world of Kakrafoon; use it to sail a small boat; use it in hand-to-hand-combat; wrap it around you to avoid the gaze of the Ravenous Bugblat Beast; it assumes that if you can't see it, it isn't there (and it's not ravenous); you can wave your towel in front of yourself off with it if it still seems to have immense psychological value. For some time, the strag discovers that a hitch hiker has his towel, and he is also in possession of a toothbrush, a compass, a map, a ball of string, a gnat spray, and a small bottle of strag. Furthermore, the strag will then happily trade you any other items that the hitch hiker might have. The hitch hiker thinks that any man who can hitch hike is clearly a man to be reckoned with.`</p>`

[Run code snippet](#)[Copy snippet to answer](#)[Hide results](#)[Full page](#)

The Hitch Hiker's Guide to the Galaxy has a few things to say on the subject of ...

answered Mar 16 '17 at 12:16



[Sawan mishra](#)

299 1 3 9

I love this solution. Just an extra tweak to `:before right: 1em; background-color: white;` so that the ellipsis stays inside the box. – [Hamid](#) Mar 13 at 12:48

- 4 Source : hackingui.com/front-end/... – [akmozo](#) Apr 1 at 15:27

... and demo : codepen.io/natonischuk/pen/QbGWBa – [akmozo](#) Apr 1 at 15:28

- 2 that will work, but you need to check if text is overflowing container with javascript. Otherwise those 3 dots will be there even if your text doesn't exceeds container – [nomak22](#) May 6 at 14:48

I finally found a solution to do what I want. As `p` a paragraphe and `article` the wrapper. If you want to apply ellipsis to `p` depending on `article` height (which also depends on window height), you need to get the height of the `article`, the line-height of the `p` and then `articleHeight/lineHeight` to find the number of `line-clamp` that can be added dynamically then.

The only thing is the `line-height` should be declared in the css file.

Check the following code. If you change the height of the window, the `line-clamp` will change. Can be great to create a plug-in aiming to do that.

[jsfiddle](#)

```
function lineclamp() {
  var lineheight = parseFloat($('p').css('line-height'));
  var articleheight = $('article').height();
  var calc = parseInt(articleheight/lineheight);
  $('p').css({'-webkit-line-clamp': calc});
}

$(document).ready(function() {
  lineclamp();
});
```

```
$( window ).resize(function() {  
    lineclamp();  
});  
  
article {  
    height:60%;  
    background:red;  
    position:absolute;  
}  
  
p {  
    margin:0;  
    line-height:120%;  
    display: -webkit-box;  
    -webkit-box-orient: vertical;  
    overflow: hidden;  
    text-overflow: ellipsis;  
}  
  
<script src="https://ajax.googleapis.c  
<article>  
    <p>Lorem ipsum dolor sit amet,  
    lacinia a justo sed, porttitor vulputa  
    nisl. Duis suscipit volutpat sollicitu  
    mollis. Pellentesque habitant morbi tr  
    turpis egestas. Nulla sollicitudin sap  
    eu sem aliquam tempus nec ut leo. Quis  
    massa ut sem egestas imperdiet. Sed sc  
    placerat lobortis ante, id ultrices ip  
    gravida ipsum nec sapien pellentesque  
    egestas. Nullam scelerisque purus inte  
    accumsan ut posuere eu, vehicula conse  
    imperdiet vel in mi. Cras leo orci, fe  
    libero. Phasellus a volutpat diam. Ut  
    vitae. Suspendisse quis lorem facilisi  
    varius nulla et enim malesuada fringil  
    adipiscing elit. Quisque lorem ligula,  
    non feugiat risus. Sed vitae urna nisl  
    massa elementum massa condimentum moll  
    et netus et malesuada fames ac turpis  
    dapibus. Pellentesque sed nisl eu sem  
    aliquam placerat. Fusce a massa ut sem  
    egestas malesuada. Quisque placerat lo  
</article>
```

[Run code snippet](#)[Copy snippet to answer](#)[Hide results](#)[Full page](#)

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Quisque lorem ligula, lacinia
a justo sed, porttitor...

answered Dec 14 '16 at 13:45



Bruno Landowski

660 1 7 16

This [man](#) have the best solution. Only CSS:

```
.multiline-ellipsis {
  display: block;
  display: -webkit-box;
  max-width: 400px;
  height: 109.2px;
  margin: 0 auto;
  font-size: 26px;
  line-height: 1.4;
  -webkit-line-clamp: 3;
  -webkit-box-orient: vertical;
  overflow: hidden;
  text-overflow: ellipsis;
}
```

edited Jul 24 '17 at 11:56



Bugs

4,149 9 25 36

answered Apr 3 '17 at 13:50



Lukasz Kurzynski

87 1 1

-
- 4 It didn't work for me. – [cmoran92](#) Aug 23 '17 at 10:40
-
- 1 This is a non standard feature that is no longer supported. – [Wojciech Maj](#) Dec 29 '17 at 13:06
-
- 1 line-clamp only works in chrome caniuse.com/#feat=css-line-clamp – [sij_a](#) Feb 23 at 11:46
-

Unfortunately no with current state of affairs in CSS.

Ellipsis rendering has prerequisite `white-space: nowrap` that effectively means: ellipsis are drawn on single line text containers only.

answered Oct 11 '15 at 3:34



c-smile

20.2k 4 41 66

-
- 1 Also haven't found one solution in CSS

that works well. It's all nacky and ugly. —
[peresisUser](#) Feb 4 '16 at 13:25

Just increase the **`-webkit-line-clamp: 4;`** to increase the number of lines

```
p {  
  display: block;  
  display: -webkit-box;  
  max-width: 200px;  
  -webkit-line-clamp: 4;  
  -webkit-box-orient: vertical;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

<p>Lorem ipsum dolor sit amet, novum n
conclusionemque. Ne qui atomorum peric
nulla pro ne, ponderum invidunt eu duc
ad fastidii consequat definitionem.**</p>**

Run code snippet

Copy snippet to answer

[Expand](#)

[snippet](#)

Edit 02/05/2018

Line clamp is a proprietary and undocumented CSS (webkit) :
<https://caniuse.com/#feat=css-line-clamp>, so it currently work on only few browsers.

edited May 2 at 15:07



[Bastien Robert](#)

323 1 13

answered Dec 14 '16 at 7:46



[Jobincs](#)

144 1 7

- 2 Works well in Chrome, but not in Internet Explorer — [Detilium](#) Feb 2 at 19:57

I have just been playing around a little bit with this concept. Basically, if you are ok with potentially having a pixel or so cut off from your last character, here is a pure css and html solution:

The way this works is by absolutely positioning a div below the viewable region of a viewport. We want the div to offset up into the visible region as our content grows. If the content grows too much, our div will offset too high, so upper bound the height our content can grow.

HTML:

```
<div class="text-container">
  <span class="text-content">
    PUT YOUR TEXT HERE
  <div class="ellipsis">...</div> //
</span>
</div>
```

CSS:

```
.text-container {
  position: relative;
  display: block;
  color: #838485;
  width: 24em;
  height: calc(2em + 5px); // This is
  // little extra space is for characters th
  overflow: hidden;
  white-space: normal;
}

.text-content {
  word-break: break-all;
  position: relative;
  display: block;
  max-height: 3em; // This preven
  // much. It should be 1 line height create
}

.ellipsis {
  position: absolute;
  right: 0;
  top: calc(4em + 2px - 100%); // Offse
  // extends below the viewport, as content
  // max-height of the content
  text-align: left;
  background: white;
}
```

I have tested this in Chrome, FF, Safari, and IE 11.

You can check it out here:

<http://codepen.io/puopg/pen/vKWJwK>

You might even be able to alleviate the abrupt cut off of the character with some CSS magic.

EDIT: I guess one thing that this imposes is word-break: break-all since otherwise the content would not extend to the very end of the viewport. :(

edited Jul 12 '16 at 18:50

answered Jul 12 '16 at 18:21



puopg

185 2 13

You don't need to include `word-break: break-all` if you don't mind the floating ellipsis. At this time you have to make a choice between the lesser of those two offenses. Someday when this is supported by default with some standard CSS rule, we'll all look back and have a good laugh. — [gfullam](#) Sep 29 '17 at 21:13

user this formula:

```
overflow: hidden;
display: -webkit-box;
-webkit-box-orient: vertical;
-webkit-line-clamp: 2; //put number of
```

answered Nov 8 at 15:28



Anton Rifco Susilo

149 3 12

```
<!DOCTYPE html>
<html>
<head>
  <style>
    /* styles for '...' */
    .block-with-text {
      width: 50px;
      height: 50px;
      /* hide text if it more than 2 lines */
      overflow: hidden;
      /* for set '...' in absolute position */
      position: relative;
      /* use this value to count lines */
      line-height: 1.2em;
      /* max-height = line-height * 3 */
      max-height: 3.6em;
      /* fix problem when last visible line is cut */
      text-align: justify;
      /* place for '...' */
      margin-right: -1em;
      padding-right: 1em;
    }

    /* create the ... */
    .block-with-text:before {
      /* points in the end */
      content: '...';
      /* absolute position */
      position: absolute;
```

```

        /* set position to right
        right: 0;
        bottom: 0;
    }
    /* hide ... if we have text
    .block-with-text:after {
        /* points in the end */
        content: '';
        /* absolute position */
        position: absolute;
        /* set position to right
        right: 0;
        /* set width and height
        width: 1em;
        height: 1em;
        margin-top: 0.2em;
        /* bg color = bg color
        background: white;
    }
</style>
</head>
<body>
    a
    <div class="block-with-text">g fdsf
asdfjk jgnsdlfkgj nsldkfjgnsldkfjgn sld
dkfjnglsdfkjng lsdkfjgn sdfgsd</div>
    <p>This is a paragraph.</p>

</body>
</html>

```

answered Dec 5 '17 at 21:25



Anup Nair

11 1

I came up with my own solution for this:

```

/*this JS code puts the ellipsis (...)
*
* to use the multiline ellipsis on an
* line-height:xxx
* height:xxx (must line-height * number
* overflow:hidden
*
* and have the class js_ellipsis
* */

//do all ellipsis when jQuery loads
jQuery(document).ready(function($) {put

//redo ellipsis when window resizes
var re_ellipsis_timeout;
jQuery( window ).resize(function() {
    //timeout mechanism prevents from clearing
    clearTimeout(re_ellipsis_timeout);
    re_ellipsis_timeout = setTimeout(function() {
        finishes"); put_ellipsisses(); }, 500);
    });

//the main function
function put_ellipsisses(){
    jQuery(".js_ellipsis").each(function()

        //remember initial text to be a
        var object_data=jQuery(this).data

```

```

if(typeof object_data.oldtext != 'string')
    jQuery(this).text(object_data.text);
} else {
    object_data.oldtext = jQuery(this).text();
    jQuery(this).data('oldtext', object_data.oldtext);
}

//truncate and ellipsis
var clientHeight = this.clientHeight;
var maxturns=100; var countturn=0;
while (this.scrollHeight > clientHeight) {
    countturn++;
    jQuery(this).text(function() {
        return text.replace(/\W/g, ' ');
    });
}
});
}
}

```

edited Apr 18 '17 at 17:22



[Andrei Gheorghiu](#)

31.7k 7 42 73

answered Apr 18 '17 at 17:01



[Pierre-Verthume Larivière](#)

41 2

After many tries, I finally ended up with a mixed js / css to handle multiline and single line overflows.

CSS3 code:

```

.forcewrap { // single line ellipsis
    -ms-text-overflow: ellipsis;
    -o-text-overflow: ellipsis;
    text-overflow: ellipsis;
    overflow: hidden;
    -moz-binding: url( 'bindings.xml#ellipsis' );
    white-space: nowrap;
    display: block;
    max-width: 95%; // spare space for ellipsis
}

.forcewrap.multiline {
    line-height: 1.2em; // my line spacing
    max-height: 3.6em; // 3 lines
    white-space: normal;
}

.manual-ellipsis:after {
    content: "\02026"; // '...'
    position: absolute; // parent container
    right: 10px; // typical padding
    bottom: 10px; // same reason
    padding-left: 5px; // spare some space
    background-color: #fff; // hide text if needed
}

```

and I simply check with js code for overflows on divs, like this:

```
function handleMultilineOverflow(div) {
  // get actual element that is overf
  var element = $(div).find('a');
  // don't know why but must get scro
  if ($(element).innerHeight() < $(el
    $(element).addClass('manual-ell
  }
}
```

Usage example in html:

```
<div class="twrap">
  <h4>
    <a class="forcewrap multiline" href
  </h4>
</div>
```

answered Mar 24 '16 at 9:40



Fabian

98 1 5

May be this can help you guys. Multi Line Ellipses with tooltip hover.

<https://codepen.io/Anugraha123/pen/WOBdOb>

```
<div>
  <p class="cards-values">Lorem ipsu
  Nunc aliquet lorem commodo, semper maur
  amet leo malesuada imperdiet eu a augue
  sem. Vivamus a felis id dui aliquam tem
  </p>
  <span class="tooltip"></span>
</div>
```

edited Jul 18 '17 at 10:07



badiya

1,659 7 18

answered Jul 18 '17 at 9:21



Anugraha Acharya

49 5

Well you could use the line-clamp function in CSS3.

```
p {
  overflow: hidden;
  text-overflow: ellipsis;
  display: -webkit-box;
  line-height: 25px;
  height: 52px;
  max-height: 52px;
  font-size: 22px;
  -webkit-line-clamp: 2;
```

```
-webkit-box-orient: vertical;  
}
```

Make sure you change the settings like you're own.

answered Nov 16 '16 at 15:26



[dutchsociety](#)

263 2 12

My solution that works for me for multiline ellipsis:

```
.crd-para {  
  color: $txt-clr-main;  
  line-height: 2rem;  
  font-weight: 600;  
  margin-bottom: 1rem !important;  
  overflow: hidden;  
  
  span::after {  
    content: "...";  
    padding-left: 0.125rem;  
  }  
}
```

edited Oct 8 at 2:37



[Pang](#)

6,820 15 63 101

answered Oct 8 at 2:18



[LoveAnimesh143](#)

11 2

Pros:

- + Cross browser (IE11, Edge, Chrome, Firefox, Safari, etc.)
- + Most natural looking

Cons:

- Adds lots of extra elements to the DOM

I wasn't satisfied with any of the workarounds I had seen. Most of them use line-clamp which is currently only supported in webkit. So I played around with it until I came up with a solution. This pure javascript solution should be compatible with IE10 and greater and all modern browsers. This

is untested outside of the stackoverflow example space below.

I think this is a good solution. The one big caveat is that it creates a span for each word inside the container, which will impact layout performance, so your mileage may vary.

```
//This is designed to be run on page load
//a function and addEventListener and call it
var $container = document.querySelector('div')

//optional - show the full text on hover
$container.title = $container.textContent

$container.textContent.trim().split(' ')
//create a span for each word and append it
var newWordSpan = document.createElement('span')
newWordSpan.textContent = word;
$container.appendChild(newWordSpan);

if (newWordSpan.getBoundingClientRect().bottom > $container.getBoundingClientRect().bottom)
//it gets into this block for the bottom of the container
//get the last visible element
var containerChildNodes = $container.childNodes
var lastVisibleElement = containerChildNodes[containerChildNodes.length - 1]

//replace this final span with the last visible element
newWordSpan.textContent = '\u2026'

//if the last visible word ended with a space
//wrapped to the next line, so we need to trim it
while (lastVisibleElement.textContent.trim().length > 0)
newWordSpan.getBoundingClientRect().bottom > lastVisibleElement.getBoundingClientRect().bottom
lastVisibleElement.style.marginBottom = 10px;
lastVisibleElement.textContent = lastVisibleElement.textContent.trim();
}

//using .some() so that we can show the full text if needed
return true;
});

.multi-line-container {
border: 1px solid lightgrey;
padding: 4px;
height: 150px;
width: 300px;
}

.ellipses-container {
display: inline-flex;
flex-wrap: wrap;
justify-content: flex-start;
align-content: flex-start; /* optional - align-content: center; */
overflow: hidden;
position: relative;
}

.ellipses-container > span {
flex: 0 0 auto;
margin-right: .25em;
}
```



```
.text-body {  
  display: none;  
}  
  
<div class="multi-line-container ellipsis">  
  <div class="text-body ellipses-text">  
elit. Pellentesque luctus ut massa eget  
iaculis nec vitae turpis. Sed pharetra  
Praesent venenatis pulvinar lectus, at  
elementum faucibus bibendum vitae, eget  
condimentum imperdiet sit amet mattis  
imperdiet. Mauris tempus rutrum metus,  
potenti. Praesent et mi enim. Orci vari  
montes, nascetur ridiculus mus.  
  </div>  
</div>
```

[Run code snippet](#)[Copy snippet to answer](#)[Expand](#)[snippet](#)

edited Oct 12 at 22:17

answered Oct 12 at 21:52



Matt L

169 1 7

To bad CSS doesn't support cross-browser multiline clamping, only webkit seems to be pushing it.

You could try and use a simple Javascript ellipsis library like Ellipsify on github the source code is very clean and small so if you do need to make any additional changes it should be quite easy.

<https://github.com/Xela101/Ellipsify>

answered Jun 7 '17 at 23:47



Xela

1,674 10 23

```
p{  
  line-height: 20px;  
  width: 157px;  
  white-space: nowrap;  
  overflow: hidden;
```

```
text-overflow: ellipsis;
}
```

or we can restrict the lines by using
and height and overflow.

answered May 23 at 6:22



Nagendra Matala

71 1 3

After looking all over the internet and trying a lot of these options, the only way to make sure that it is covered correctly with support for i.e is through javascript, i created a loop function to go over post items that require multi line truncation.

*note i used JQuery, and requires your post__items class to have a fixed max-height.

```
// Loop over post items
$('.post__items').each(function(){
    var textArray = $(this).text().split(' ');
    while($(this).prop('scrollHeight') > $(this).height()){
        textArray.pop();
        $(this).text(textArray.join(' '));
    }
});
```

answered Mar 2 at 5:08



Paddy

466 5 19

If you also have multiple elements and you want a link with read more button after ellipsis, take a look on <https://stackoverflow.com/a/51418807/10104342>

If you want something like this:

Every month first 10 TB are are not charged. All other traffic... Read more

answered Jul 19 at 9:21



Diana Eftimie

1

Try this Simple way. I have also [jsfiddle](#) link here.

Here is `-webkit-line-clamp` has an important role. The value of this property decides how much lines are visible.

CSS PART

```
h3{
  text-align: center;
  overflow: hidden;
  text-overflow: ellipsis;
  margin: 0px 30px;
  display: -webkit-box;
  -webkit-line-clamp: 3;
  -webkit-box-orient: vertical;
}
```

HTML PART

`<h3>` Lorem Ipsum is simply dummy text of Ipsum has been the industry's standard printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was the first time a typesetting system using a computer. It was Letraset sheets containing Lorem Ipsum publishing software like Aldus PageMaker.

answered Jan 5 at 9:41



Ajay Gupta

1,380 12 21