

What I am looking for:

A way to style one **HALF** of a character. (In this case, half the letter being transparent)

What I have currently searched for and tried (With no luck):

- Methods for styling half of a character/letter
- Styling part of a character with CSS or JavaScript
- Apply CSS to 50% of a character

Below is an example of what I am trying to obtain.



Does a CSS or JavaScript solution exist for this or am I going to have to resort to images? I would prefer not to go the image route as this text will end up being generated dynamically.

UPDATE:

Since many have asked why I would ever want to style half of a character, this is why. My city had recently spent \$250,000 to define a new "brand" for itself. This [LOGO](#) is what they came up with. Many people have complained about the simplicity and lack of creativity and continue to do so. My goal was to come up with this [WEBSITE](#) as a joke. Type in 'Halifax' and you will see what I mean. :)

[javascript](#) [html](#) [css](#)

edited Jun 10 at 14:25

asked May 9 '14 at 16:16

 [Mathew MacLean](#)
5,140 ● 3 ● 12 ● 33

83 fixed width containers with overflow hidden. Have fun with font sizes.... – [epascarello](#) May 9 '14 at 16:18

20 You can do it with CSS if the character is drawn with SVG. [Relevant recent blog post](#) – [Pointy](#) May 9 '14 at 16:18

25 This will be a real nightmare in a real production app, did it for fun: [jsfiddle.net/9WWsd](#) – [epascarello](#) May 9 '14 at 16:26

13 I need to go to bed, but let me know how this works for you: [jsfiddle.net/BoltClock/v6M9r](#) Some of the glaring issues like duplicated content messing up C&P, and text wrapping are mitigated but it's still not perfect - highlighting before C&P sucks, for example, but for the most part, C&P actually works. – [BoltClock](#) ♦ May 9 '14 at 17:55

8 Eventually you're going to need to apply CSS separately to the [bar in the A too...](#) – [mrb](#) May 12 '14 at 17:36

Now on GitHub as a Plugin!



Feel free to fork and improve.

[Demo](#) | [Download Zip](#) | [Half-Style.com](#) (Redirects to GitHub)

- **Pure CSS** for a Single Character
- JavaScript used for automation across text or multiple characters
- Preserves Text Accessibility for screen readers for the blind or visually impaired

Part1: Basic Solution

Single Characters:

XYZA

Automated on any text:

Half-style, please.

Demo: <http://jsfiddle.net/pd9yB/817/>

This works on any dynamic text, or a single character, and is all automated. All you need to do is add a class on the target text and the rest is taken care of.

Also, the accessibility of the original text is preserved for screen readers for the blind or visually impaired.

Explanation for a single character:

Pure CSS. All you need to do is to apply `.halfStyle` class to each element that contains the character you want to be half-styled.

For each span element containing the character, you can create a data attribute, for example here `data-content="X"`, and on the pseudo element use `content: attr(data-content);` so the `.halfStyle:before` class will be dynamic and you won't need to hard code it for every instance.

Explanation for any text:

Simply add `textToHalfStyle` class to the element containing the text.

CSS:

```
.halfStyle {
  position: relative;
  display: inline-block;
  font-size: 80px; /* or any font size will work */
  color: black; /* or transparent, any color */
  overflow: hidden;
  white-space: pre; /* to preserve the spaces from collapsing */
}
.halfStyle:before {
  display: block;
  z-index: 1;
  position: absolute;
  top: 0;
  left: 0;
  width: 50%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow: hidden;
  color: #f00;
}
```

HTML

```
<p>Single Characters:</p>
<span class="halfStyle" data-content="X">X</span>
<span class="halfStyle" data-content="Y">Y</span>
<span class="halfStyle" data-content="Z">Z</span>
<span class="halfStyle" data-content="A">A</span>

<hr/>
<p>Automated:</p>

<span class="textToHalfStyle">Half-style, please.</span>
```

To make it automated, simply add `textToHalfStyle` class to the element containing the text.

jQuery for automated mode:

```
jQuery(function($) {
  var text, chars, $el, i, output;

  // Iterate over all class occurrences
  $('textToHalfStyle').each(function(idx, el) {
```

```
$el = $(el);
text = $el.text();
chars = text.split('');

// Set the screen-reader text
$el.html('<span style="position: absolute !important; clip: rect(1px 1px 1px 1px); clip: rect(1px, 1px, 1px, 1px);">' + text + '</span>');

// Reset output for appending
output = '';

// Iterate over all chars in the text
for (i = 0; i < chars.length; i++) {
    // Create a styled element for each character and append to container
    output += '<span aria-hidden="true" class="halfStyle" data-content="' +
chars[i] + '>' + chars[i] + '</span>';
}

// Write to DOM only once
$el.append(output);
});
});
```

Demo: <http://jsfiddle.net/pd9yB/819/>

Part2: Advanced solution - Independent left and right parts

Single Characters:

XYZA

Automated:

Half-style, please.

With this solution you can style left and right parts, individually and independently.

Everything is the same, only more advanced CSS does the magic.

Demo: <http://jsfiddle.net/pd9yB/819/>

```
.halfStyle {
    position: relative;
    display: inline-block;
    font-size: 80px; /* or any font size will work */
    color: transparent; /* hide the base character */
    overflow: hidden;
    white-space: pre; /* to preserve the spaces from collapsing */
}
.halfStyle:before { /* creates the left part */
    display: block;
    z-index: 1;
    position: absolute;
    top: 0;
    width: 50%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    overflow: hidden;
    pointer-events: none; /* so the base char is selectable by mouse */
    color: #f00; /* for demo purposes */
    text-shadow: 2px -2px 0px #af0; /* for demo purposes */
}
.halfStyle:after { /* creates the right part */
    display: block;
    direction: rtl; /* very important, will make the width to start from right */
    position: absolute;
    z-index: 2;
    top: 0;
    left: 50%;
    width: 50%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    overflow: hidden;
    pointer-events: none; /* so the base char is selectable by mouse */
    color: #000; /* for demo purposes */
    text-shadow: 2px 2px 0px #0af; /* for demo purposes */
}
```

Part3: Mix-Match and Improve

Now that we know what is possible, let's create some variations.

-Horizontal Half Parts

Single Characters:



The image shows the characters 'X', 'Y', 'Z', and 'A' in a horizontal half-style font. Each character is split vertically down the middle. The left half of each character is blue, and the right half is red. The characters are bold and have a slight shadow effect.

Automated on any text:



The image shows the text 'Half-style, please.' in a horizontal half-style font. The text is split vertically down the middle, with the left half being blue and the right half being red. The font is bold and has a slight shadow effect.

Demo

```
.halfStyle {
  position:relative;
  display:inline-block;
  font-size:80px; /* or any font size will work */
  color: transparent; /* hide the base character */
  overflow:hidden;
  white-space: pre; /* to preserve the spaces from collapsing */
}
.halfStyle:before { /* creates the top part */
  display:block;
  z-index:2;
  position:absolute;
  top:0;
  height: 50%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
  color: #f00; /* for demo purposes */
  text-shadow: 2px -2px 0px #af0; /* for demo purposes */
}
.halfStyle:after { /* creates the bottom part */
  display:block;
  position:absolute;
  z-index:1;
  top:0;
  height: 100%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
  color: #000; /* for demo purposes */
  text-shadow: 2px 2px 0px #0af; /* for demo purposes */
}
```

-Vertical 1/3 Parts

Single Characters:



The image shows the characters 'X', 'Y', 'Z', and 'A' in a vertical 1/3 parts font. Each character is split into three vertical sections. The left section is blue, the middle section is red, and the right section is green. The characters are bold and have a slight shadow effect.

Automated on any text:



The image shows the text 'Half-style, please.' in a vertical 1/3 parts font. The text is split into three vertical sections: blue on the left, red in the middle, and green on the right. The font is bold and has a slight shadow effect.

Demo

```
.halfStyle { /* base char and also the right 1/3 */
  position:relative;
  display:inline-block;
  font-size:80px; /* or any font size will work */
  color: transparent; /* hide the base character */
  overflow:hidden;
  white-space: pre; /* to preserve the spaces from collapsing */
  color: #f0f; /* for demo purposes */
  text-shadow: 2px 2px 0px #0af; /* for demo purposes */
}
.halfStyle:before { /* creates the left 1/3 */
  display:block;
  z-index:2;
  position:absolute;
  top:0;
  width: 33.33%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
  color: #f00; /* for demo purposes */
  text-shadow: 2px -2px 0px #af0; /* for demo purposes */
}
.halfStyle:after { /* creates the middle 1/3 */
  display:block;
  z-index:1;
  position:absolute;
  top:0;
  width: 66.66%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
  color: #000; /* for demo purposes */
  text-shadow: 2px 2px 0px #af0; /* for demo purposes */
}
```

-Horizontal 1/3 Parts

Single Characters:

XYZA

Automated on any text:

Half-style, please.

Demo

```
.halfStyle { /* base char and also the bottom 1/3 */
  position:relative;
  display:inline-block;
  font-size:80px; /* or any font size will work */
  color: transparent;
  overflow:hidden;
  white-space: pre; /* to preserve the spaces from collapsing */
  color: #f0f;
  text-shadow: 2px 2px 0px #0af; /* for demo purposes */
}
.halfStyle:before { /* creates the top 1/3 */
  display:block;
  z-index:2;
  position:absolute;
  top:0;
  height: 33.33%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
  color: #f00; /* for demo purposes */
  text-shadow: 2px -2px 0px #fa0; /* for demo purposes */
}
.halfStyle:after { /* creates the middle 1/3 */
  display:block;
  position:absolute;
  z-index:1;
  top:0;
  height: 66.66%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  pointer-events: none; /* so the base char is selectable by mouse */
}
```

```
color: #000; /* for demo purposes */
text-shadow: 2px 2px 0px #af0; /* for demo purposes */
}
```

-HalfStyle Improvement By @KevinGranger



DEMO

```
body{
  background-color: black;
}

.textToHalfStyle{
  display:block;
  margin: 200px 0 0 0;
  text-align:center;
}

.halfStyle {
  font-family: 'Libre Baskerville', serif;
  position:relative;
  display:inline-block;
  width:1;
  font-size:70px;
  color: black;
  overflow:hidden;
  white-space: pre;
  text-shadow: 1px 2px 0 white;
}

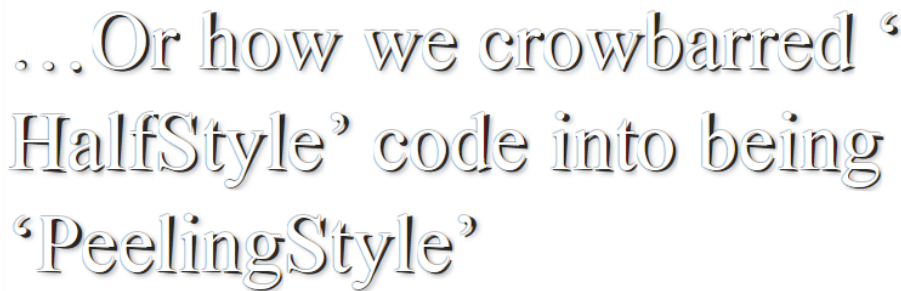
.halfStyle:before {
  display:block;
  z-index:1;
  position:absolute;
  top:0;
  width: 50%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow:hidden;
  color: white;
}
```

-PeelingStyle improvement of HalfStyle by @SamTremaine

Single Characters:



Automated on any text:



```
.halfStyle {
  position: relative;
  display: inline-block;
  font-size: 68px;
  color: rgba(0, 0, 0, 0.8);
  overflow: hidden;
  white-space: pre;
  transform: rotate(4deg);
  text-shadow: 2px 1px 3px rgba(0, 0, 0, 0.3);
}
.halfStyle:before { /* creates the left part */
  display: block;
  z-index: 1;
  position: absolute;
  top: -0.5px;
  left: -3px;
  width: 100%;
  content: attr(data-content);
  overflow: hidden;
  pointer-events: none;
  color: #FFF;
  transform: rotate(-4deg);
  text-shadow: 0px 0px 1px #000;
}
```

Part4: Ready for Production

Customized different Half-Style style-sets can be used on desired elements on the same page. You can define multiple style-sets and tell the plugin which one to use.

The plugin uses data attribute `data-halfstyle="[-CustomClassName-]"` on the target `.textToHalfStyle` elements and makes all the necessary changes automatically.

So, simply on the element containing the text add `textToHalfStyle` class and data attribute `data-halfstyle="[-CustomClassName-]"`. The plugin will do the rest of the job.

Half-style, please.

Half-style, please.

Half-style, please.

Half-style, please.

Demo of Multiple Half-Styles on the same page.

```
jQuery(function($) {
  var halfstyle_text, halfstyle_chars, $halfstyle_el, halfstyle_i, halfstyle_output,
  halfstyle_style;

  // Iterate over all class occurrences
  $('textToHalfStyle').each(function(idx, halfstyle_el) {
    $halfstyle_el = $(halfstyle_el);
    halfstyle_style = $halfstyle_el.data('halfstyle');
    halfstyle_text = $halfstyle_el.text();
    halfstyle_chars = halfstyle_text.split('');

    // Set the screen-reader text
    $halfstyle_el.html('<span style="position: absolute !important;clip: rect(1px 1px 1px 1px);clip: rect(1px, 1px, 1px, 1px);">' + halfstyle_text + '</span>');

    // Reset output for appending
    halfstyle_output = '';

    // Iterate over all chars in the text
    for (halfstyle_i = 0; halfstyle_i < halfstyle_chars.length; halfstyle_i++) {
```

```

        // Create a styled element for each character and append to container
        halfstyle_output += '<span aria-hidden="true" class="halfStyle ' +
halfstyle_style + '" data-content="' + halfstyle_chars[halfstyle_i] + '>' +
halfstyle_chars[halfstyle_i] + '</span>';
    }

    // Write to DOM only once
    $halfstyle_el.append(halfstyle_output);
});
});

```

Also the CSS style-sets' class definitions match the `[-CustomClassName-]` part mentioned above and is chained to `.halfStyle`, so we will have `.halfStyle.[-CustomClassName-]`

```

/* start half-style hs-base */
.halfStyle.hs-base {
    position:relative;
    display:inline-block;
    font-size:80px; /* or any font size will work */
    overflow:hidden;
    white-space: pre; /* to preserve the spaces from collapsing */
    color: #000; /* for demo purposes */
}
.halfStyle.hs-base:before {
    display:block;
    z-index:1;
    position:absolute;
    top:0;
    width: 50%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    pointer-events: none; /* so the base char is selectable by mouse */
    overflow:hidden;
    color: #f00; /* for demo purposes */
}
/* end half-style hs-base */

/* start half-style hs-horizontal-third */
.halfStyle.hs-horizontal-third { /* base char and also the bottom 1/3 */
    position:relative;
    display:inline-block;
    font-size:80px; /* or any font size will work */
    color: transparent;
    overflow:hidden;
    white-space: pre; /* to preserve the spaces from collapsing */
    color: #f0f;
    text-shadow: 2px 2px 0px #0af; /* for demo purposes */
}
.halfStyle.hs-horizontal-third:before { /* creates the top 1/3 */
    display:block;
    z-index:2;
    position:absolute;
    top:0;
    height: 33.33%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    overflow:hidden;
    pointer-events: none; /* so the base char is selectable by mouse */
    color: #f00; /* for demo purposes */
    text-shadow: 2px -2px 0px #fa0; /* for demo purposes */
}
.halfStyle.hs-horizontal-third:after { /* creates the middle 1/3 */
    display:block;
    position:absolute;
    z-index:1;
    top:0;
    height: 66.66%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    overflow:hidden;
    pointer-events: none; /* so the base char is selectable by mouse */
    color: #000; /* for demo purposes */
    text-shadow: 2px 2px 0px #af0; /* for demo purposes */
}
/* end half-style hs-horizontal-third */

/* start half-style hs-PeelingStyle, by user SamTremaine on Stackoverflow.com */
.halfStyle.hs-PeelingStyle {
    position: relative;
    display: inline-block;
    font-size: 68px;
    color: rgba(0, 0, 0, 0.8);
    overflow: hidden;
    white-space: pre;
    transform: rotate(4deg);
    text-shadow: 2px 1px 3px rgba(0, 0, 0, 0.3);
}
.halfStyle.hs-PeelingStyle:before { /* creates the left part */
    display: block;
    z-index: 1;
    position: absolute;
    top: -0.5px;
    left: -3px;
    width: 100%;
    content: attr(data-content);
    overflow: hidden;
    pointer-events: none;
    color: #FFF;
    transform: rotate(-4deg);
}

```



```

    text-shadow: 0px 0px 1px #000;
}
/* end half-style hs-PeelingStyle */

/* start half-style hs-KevinGranger, by user KevinGranger on StackOverflow.com*/
.textToHalfStyle.hs-KevinGranger {
    display: block;
    margin: 200px 0 0 0;
    text-align: center;
}

.halfStyle.hs-KevinGranger {
    font-family: 'Libre Baskerville', serif;
    position: relative;
    display: inline-block;
    width: 1;
    font-size: 70px;
    color: black;
    overflow: hidden;
    white-space: pre;
    text-shadow: 1px 2px 0 white;
}
.halfStyle.hs-KevinGranger:before {
    display: block;
    z-index: 1;
    position: absolute;
    top: 0;
    width: 50%;
    content: attr(data-content); /* dynamic content for the pseudo element */
    overflow: hidden;
    color: white;
}
}
/* end half-style hs-KevinGranger

```

HTML:

```

<p>
  <span class="textToHalfStyle" data-halfstyle="hs-base">Half-style, please.</span>
</p>
<p>
  <span class="textToHalfStyle" data-halfstyle="hs-horizontal-third">Half-style, please.
</span>
</p>
<p>
  <span class="textToHalfStyle" data-halfstyle="hs-PeelingStyle">Half-style, please.
</span>
</p>
<p style="background-color:#000;">
  <span class="textToHalfStyle" data-halfstyle="hs-KevinGranger">Half-style, please.
</span>
</p>

```

Demo of Multiple Half-Styles on the same page.

edited May 16 '14 at 17:13

answered May 9 '14 at 16:42



Arbel

13.6k ● 2 ● 8 ● 18

- 8 This one works really well but removes spaces if you put something like a sentence or phrase. – [mttdbrd](#) May 9 '14 at 19:43
- 9 @mttdbr and ejegg, Fixed. Just needed to apply `white-space:pre;` to `.halfStyle` – [Arbel](#) May 9 '14 at 20:53
- 22 Cool, but this doesn't play nice with people wanting to select text. You can add `pointer-events: none;` to make it work, so that a click (using the first example) on the red part of the text is passed through to the hidden black part of the text behind it. – [hvd](#) May 10 '14 at 9:26
- 219 The amazing thing about SO is that no matter how silly your problem is, someone will find a solution. – [jwg](#) May 14 '14 at 8:04
- 11 My eyes hurt. I hope it won't be used much. Anyway, nice answer. – [KBart](#) May 14 '14 at 10:51



Splitchar



I've just finished developing the plugin and it is available for everyone to use! Hope you will enjoy it.

View Project on [GitHub](#) - View Project [Website](#). (so you can see all the split styles)

Usage

First of all, make sure you have the `jQuery` library is included. The best way to get the latest jQuery version is to update your head tag with:

```
<script src="http://code.jquery.com/jquery-latest.min.js"></script>
```

After downloading the files, make sure you include them in your project:

```
<link rel="stylesheet" type="text/css" href="css/splitchar.css">
<script type="text/javascript" src="js/splitchar.js"></script>
```

Markup

All you have to do is to assign the class `splitchar`, followed by the desired style to the element wrapping your text. e.g

```
<h1 class="splitchar horizontal">Splitchar</h1>
```

After all this is done, just make sure you call the jQuery function in your document ready file like this:

```
$(".splitchar").splitchar();
```

Customizing

In order to make the text look exactly as you want it to, all you have to do is apply your design like this:

```
.horizontal { /* Base CSS - e.g font-size */ }
.horizontal:before { /* CSS for the left half */ }
.horizontal:after { /* CSS for the right half */ }
```

That's it! Now you have the `Splitchar` plugin all set. More info about it at emisfera.github.io/Splitchar.

edited May 12 '14 at 17:34

answered May 9 '14 at 16:32



Razvan Balosin

3,497 1 4 19

5 +1 This is nice and I like that it's CSS only. – [Dryden Long](#) May 9 '14 at 16:34

2 This has problems with text wrapping which is exhibited even in the latest fiddle. When one character wraps, it essentially splits into two. Should be a trivial fix, though. – [BoltClock](#) ♦ May 9 '14 at 17:15

1 @MathewMacLean Check this out: [link](#) :) – [Razvan Balosin](#) May 10 '14 at 9:59

9 Don't depend on jquery-latest.min.js, it can make your sites break without warning if jQuery is updated and the plugin does not work with the newer one. Instead: use a specific version and check compatibility when updating. – [Niels Bom](#) May 13 '14 at 13:39

4 posting actual code on stackoverflow wud have been better. But Thanks :) – [STEEL](#) May 14 '14 at 4:26

Yes, you can do this with only one character and only CSS.

Webkit (and Chrome) only, though:

<http://jsbin.com/rexoyice/1/>

HTML:

```
<h1>X</h1>
```

CSS:

```
h1 {
  display: inline-block;
  font-family: helvetica, arial, sans-serif;
  font-weight: bold;
  font-size: 300px;
  background: linear-gradient(to right, #7db9e8 50%, #1e5799 50%);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
}
```

Visually, all the examples that use two characters (be it via JS, CSS pseudo elements, or just HTML) look fine, but note that that all adds content to the DOM which may cause accessibility--as well as text selection/cut/paste issues.

edited May 13 '14 at 3:49

answered May 9 '14 at 16:42



DA.

18.7k ● 29 ● 93 ● 159

2 This is interesting, although unfortunately only webkit. – [Mathew MacLean](#) May 9 '14 at 16:45

18 @MathewMacLean our jobs would be so much easier if only IE would die and Firefox started using Webkit. :) – [DA](#). May 9 '14 at 16:46

5 @DA Chrome doesn't use webkit anymore: wired.com/2013/04/blink – [Matt Harrison](#) May 12 '14 at 19:43

28 WebKit has a history of rendering bugs that are almost IE6/IE7 levels of bizarre (you could even say that Safari and Chrome are the IE6 of the modern web), and behaving in ways that deviate from the standard for no particular reason. IE has been much better since version 9, so while the ancient versions ought to die already, I don't see any reason for the hate for its recent versions. And I certainly don't see why people support the idea of a WebKit/Blink monoculture (the comments here are probably in jest, but I've heard of people who seriously believe in it). – [BoltClock](#) ♦ May 13 '14 at 5:38

1 That being said, `background-clip: text` is super awesome and they should consider it (or something similar like `text-decoration-background`) for a level 4 module. – [BoltClock](#) ♦ May 13 '14 at 5:40

Two is better than one.

JSFiddle DEMO

We'll do it using just CSS pseudo selectors!

This technique will work with dynamically generated content and different font sizes and widths.

HTML:

```
<div class='split-color'>Two is better than one.</div>
```

CSS:

```
.split-color > span {
  white-space: pre-line;
  position: relative;
  color: #409FBF;
}

.split-color > span:before {
  content: attr(data-content);
  pointer-events: none; /* Prevents events from targeting pseudo-element */
  position: absolute;
  overflow: hidden;
  color: #264A73;
  width: 50%;
  z-index: 1;
}
```

To wrap the dynamically generated string, you could use a function like this:

```
// Wrap each letter in a span tag and return an HTML string
```

```
// that can be used to replace the original text
function wrapString(str) {
  var output = [];
  str.split('').forEach(function(letter) {
    var wrapper = document.createElement('span');
    wrapper.dataset.content = wrapper.innerHTML = letter;

    output.push(wrapper.outerHTML);
  });

  return output.join('');
}

// Replace the original text with the split-color text
window.onload = function() {
  var el = document.querySelector('.split-color'),
      txt = el.innerHTML;

  el.innerHTML = wrapString(txt);
}
```

edited May 11 '14 at 19:54



Peter Mortensen

9,106 ● 10 ● 63 ● 98

answered May 9 '14 at 16:33



wvandaal

2,183 ● 1 ● 6 ● 23

This is neat, but the only issue is the content will be dynamic. – Mathew MacLean May 9 '14 at 16:35

Results vary depending on the font used. Plus calculating the width seems like a problem. – j08691 May 9 '14 at 16:35

1 @MathewMacLean you can write a simple loop function in JS to accomplish the wrapping. I'm adding it to my answer now. – wvandaal May 9 '14 at 16:55

1 @MathewMacLean Where is the text coming from? wvandaal is right, you can wrap the text yourself. – mttddbrd May 9 '14 at 16:55

1 @MathewMacLean here's an example: jsfiddle.net/CL82F/5 – wvandaal May 9 '14 at 17:24

It may be irrelevant, maybe not, but sometime ago, I created a jQuery function that does the same thing, but horizontally.

I called it "Strippex" For 'stripe'+ 'text', demo : <http://cdpn.io/FclBg>

I'm not saying this is the solution of any problems, but I already tried to apply css to half of a character, but horizontally, So the idea is the same, the realisation may be horrible, but it works.

Ah, and the most important, I had fun creating it !

edited May 15 '14 at 21:58



GenericJam

315 ● 2 ● 5 ● 15

answered May 13 '14 at 11:05



LukyVj

645 ● 5 ● 13

1 @Luky Vj: Is this account yours? You might want to consolidate all your posts into a single account so you don't run into roadblocks trying to edit your own posts. – BoltClock ♦ May 13 '14 at 12:08

Yeah, in fact, I firstly posted with my old first account.. And I had to add an image, and I wasn't popular enough to post my image.. But you're right, I will fix it as soon as possible ! – LukyVj May 13 '14 at 12:45

@LukyVj: You can merge your accounts by following the instructions here: stackoverflow.com/help/merging-accounts – BoltClock ♦ May 14 '14 at 2:38

1 @LukyVj I updated your function by adding `pointer-events:none` to `&:nth-child(2) - &:nth-child(5)`. This makes it so the text can only be highlighted once and you only get one copy of it. You can see it here: codepen.io/anon/pen/upLaj – Mathew MacLean May 14 '14 at 16:59

Here an ugly implementation in canvas. I tried this solution, but the results are worse than I expected, so here it is anyway.

Example Text

<http://jsfiddle.net/kLXmL/2/>

```
<div>Example Text</div>

$(function(){
  $("div").each(function(){
    var CHARS = $(this).text().split('');
    $(this).html("");
    $.each(CHARS,function(index, char){
      var canvas = $("<canvas />")
        .css("width", "40px")
        .css("height", "40px")
        .get(0);
      $("div").append(canvas);
      var ctx = canvas.getContext("2d");
      var gradient = ctx.createLinearGradient(0, 0, 130, 0);
      gradient.addColorStop("0", "blue");
      gradient.addColorStop("0.5", "blue");
      gradient.addColorStop("0.51", "red");
      gradient.addColorStop("1.0", "red");
      ctx.font = '130pt Calibri';
      ctx.fillStyle = gradient;
      ctx.fillText(char, 10, 130);
    });
  });
});
```

edited May 11 '14 at 19:59



Peter Mortensen

9,106 ● 10 ● 63 ● 98

answered May 9 '14 at 19:33



HaSuKrOnOs

999 ● 4 ● 16

BTW, you can use 0.5 for the red colour stop, too. – [Toothbrush](#) Jun 9 '14 at 15:39

Closest I can get:

```
$(function(){
  $('span').width($('span').width()/2);
  $('span:nth-child(2)').css('text-indent', -$('span').width());
});
```

CSS:

```
body{
  font-family: arial;
}
span{
  display: inline-block;
  overflow: hidden;
}
span:nth-child(2){
  color: red;
}
```

HTML:

```
<span>X</span><span>X</span>
```

Demo: <http://jsfiddle.net/9wxfY/2/>

Heres a version that just uses one span: <http://jsfiddle.net/9wxfY/4/>

edited May 9 '14 at 16:34

answered May 9 '14 at 16:28



Prisoner

16k ● 6 ● 43 ● 79

1 `$('span').width()` just returns the width of the first span it finds; it'd have to be something you did for each pair. Which gives me an idea ... – [Pointy](#) May 9 '14 at 16:30

This is quite similar to epascarello's example found at jsfiddle.net/9WWsd. As I told him, Your example is a step in the right direction, but, it would be a nightmare to use on a larger scale. – [Mathew MacLean](#) May 9 '14 at 16:30

@MathewMacLean, I didn't see that. Why would it be a nightmare? How about this: jsfiddle.net/9wxfY/4 – [Prisoner](#) May 9 '14 at 16:33

When you go to implement more than one character it causes problems. – [Mathew MacLean](#) May 9 '14 at 16:34

1 @Pointy, yeah I know, it's just an example. – [Prisoner](#) May 9 '14 at 16:36

Nice trick, [CSS](#) is everything :)




I just played with [@Arbel's](#) solution:

```
<span class="textToHalfStyle">Dr. Jekyll and M. Hide</span>

<style>
  body{
    background-color: black;
  }
  .textToHalfStyle{
    display: block;
    margin: 200px 0 0 0;
    text-align:center;
  }
  .halfStyle {
    font-family: 'Libre Baskerville', serif;
    position: relative;
    display: inline-block;
    width: 1;
    font-size: 70px;
    color: black;
    overflow: hidden;
    white-space: pre;
    text-shadow: 1px 2px 0 white;
  }
  .halfStyle:before {
    display: block;
    z-index: 1;
    position: absolute;
    top: 0;
    width: 50%;
    content: attr(data-content); /* Dynamic content for the pseudo element */
    overflow: hidden;
    color: white;
  }
</style>
```


http://jsfiddle.net/Kevin_Granger/pd9yB/327/

edited Nov 26 '14 at 22:58

 [Peter Mortensen](#)

9,106 ● 10 ● 63 ● 98

answered May 12 '14 at 20:32

 [Shipow](#)

1,151 ● 1 ● 11 ● 23

- 3 Linking to a jsFiddle demo is fine, but please include the relevant CSS here as well. – [p.s.w.g](#) May 12 '14 at 21:33
- Just played with @Arbel solution* What are the differences with Arbel's solution? It's hard to see if you only copy-pasted some code or improved it. – [A.L](#) May 13 '14 at 13:40
- 1 I think it's pretty easy to see – [Jeremy Cook](#) May 13 '14 at 18:01
- 1 @KevinGranger This is cool. I added this to the main answer with references to you! Also I have added three more base solutions which will give you more opportunities to get creative. – [Arbel](#) May 14 '14 at 4:50
- 1 Something off-topic, but... This is a very good answer to a quite very technical and niche question. Do you really think, that people, who read this question and yours and other answers really need a link to Wikipedia, explaining, what CSS is? :] – [trejder](#) Dec 10 '14 at 10:11

Another CSS-only solution (though data-attribute is needed if you don't want to write letter-specific CSS). This one works more across the board (Tested IE 9/10, Chrome latest & FF latest)

<http://jsfiddle.net/pkML2/>

HTML

```
<span data-char="X">X</span>
```

CSS

```
span {
  position: relative;
  color: rgba(50,50,200,0.5);
}

span:before {
  content: attr(data-char);
  position: absolute;
  width: 50%;
  overflow: hidden;
  color: rgb(50,50,200);
}
```

answered May 12 '14 at 18:44



MStrutt

469 ● 2 ● 6

1 The accepted answer uses this method as does one or two other answers.. – [Mathew MacLean](#) May 12 '14 at 18:48

1 I know the accepted answer uses this technique at it's core but I was attempting to give a more minimal answer with only what is required to achieve the result asked for. It certainly isn't as comprehensive as some of the other answers, but is very light-weight. – [MStrutt](#) May 12 '14 at 21:47

Excellent. Worth at least \$25. I guess the government contact relationship was worth the rest! – [maxweber](#) May 20 '14 at 16:56

If you are interested in this, then Lucas Bebbler's Glitch is a very similar and super cool effect:



Created using a simple SASS Mixin such as

```
.example-one {
  font-size: 100px;
  @include textGlitch("example-one", 17, white, black, red, blue, 450, 115);
}
```

More details at [Chris Coyer's CSS Tricks](#) and [Lucas Bebbler's Codepen page](#)

answered Oct 1 '14 at 12:39



Ruskin

2,088 ● 1 ● 12 ● 35

1 Pretty cool! Thanks for the share! – [Mathew MacLean](#) Oct 2 '14 at 12:18

```
.halfStyle {
  position: relative;
  display: inline-block;
  font-size: 68px; /* or any font size will work */
  color: rgba(0,0,0,0.8); /* or transparent, any color */
  overflow: hidden;
  white-space: pre; /* to preserve the spaces from collapsing */
  transform: rotate(4deg);
  -webkit-transform: rotate(4deg);
  text-shadow: 2px 1px 3px rgba(0,0,0,0.3);
}

.halfStyle:before {
  display: block;
  z-index: 1;
  position: absolute;
  top: -0.5px;
  left: -3px;
  width: 100%;
  content: attr(data-content); /* dynamic content for the pseudo element */
  overflow: hidden;
  color: white;
  transform: rotate(-4deg);
  -webkit-transform: rotate(-4deg);
  text-shadow: 0 1px black;
}
```

<http://experimental.samtremaine.co.uk/half-style/>

You can crowbar this code into doing all sorts of interesting things - this is just one implementation my associate and I came up with last night.

answered May 13 '14 at 9:26



[Sam Tremain](#)

171 ● 2

1 Ohh wow that's actually really cool looking! – [Mathew MacLean](#) May 13 '14 at 9:35

1 @SamTremain This is really cool! I added this to the main answer with references to you! Also I have added three more base solutions which will give you more opportunities to get creative. – [Arbel](#) May 14 '14 at 4:51

@Arbel You spelt SamTremain wrong in your answer. Thought I would let you know, – [Mathew MacLean](#) May 14 '14 at 13:13

@MathewMacLean Fixed the spelling. Thanks. – [Arbel](#) May 14 '14 at 14:29

I am not sure how elegant this solution is, but it cuts everything exactly in half:

<http://jsfiddle.net/9wxFY/11/>

Otherwise, I have created a nice solution for you... All you need to do is have this for your HTML:

Take a look at this: <http://jsfiddle.net/9wxFY/8/>

As for the CSS, it is very limited... You only need to apply it to `:nth-child(even)`

CSS

```
.half-color span{
  font-size: 2em;
  display: inline-block;
  overflow: hidden;
}
.half-color span:nth-child(even){
  color: red;
}
```

HTML

```
<div class="half-color">This is a sentence</div>
```

jQuery

```
$(function(){
  var $hc = $('.half-color');
  var str = $hc.text();
  $hc.html("");

  var i = 0;
  var chars;
  var dupText;

  while(i < str.length){
    chars = str[i];
    dupText = "<span>" + chars + "</span>";

    $hc.append(dupText);
    $hc.append(dupText);

    i++;
  }

  $('.half-color span:nth-child(odd)').width($('.half-color span:first-child').width()/2);
  $('.half-color span:nth-child(even)').css('text-indent', -$('.half-color span:first-child').width());
});
```

edited May 11 '14 at 19:57



[Peter Mortensen](#)

9,106 ● 10 ● 63 ● 98

answered May 9 '14 at 16:39



[Adjit](#)

4,323 ● 2 ● 13 ● 36

This has the same issues that the others have though. This starts getting messy when you try to do it with more than one character. – [Mathew MacLean](#) May 9 '14 at 16:41

@MathewMacLean ahhh, now I see. So what would you want? every character to be half colored? or the sentence as a whole to be half colored? – [Adjit](#) May 9 '14 at 16:43

Each individual character. – [Mathew MacLean](#) May 9 '14 at 16:46

@MathewMacLean made a nice solution for you! – [Adjit](#) May 9 '14 at 17:10

This is very nice, however it is not accurate to the "half". look at this fiddle: jsfiddle.net/9wxfY/9 – [Mathew MacLean](#) May 9 '14 at 17:14

A nice WebKit-only solution that takes advantage of the `background-clip: text` support: http://jsfiddle.net/sandro_paganotti/wLkVt/

```
span{
  font-size: 100px;
  background: linear-gradient(to right, black, black 50%, grey 50%, grey);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
}
```

answered May 9 '14 at 17:50



[Sandro Paganotti](#)

1,164 ● 5 ● 10

Already suggested in DA's answer. – [BoltClock](#) ♦ May 9 '14 at 17:51

Sorry @da, I didn't noticed your answer – [Sandro Paganotti](#) May 9 '14 at 17:53

FWIW, here's my take on this doing it only with CSS: <http://codepen.io/ricardozea/pen/uFbts/>

Several notes:

- The main reason I did this was to test myself and see if I was able to accomplish styling half of a character while actually providing a meaningful answer to the OP.
- I am aware that this is not an ideal or the most scalable solution and the solutions proposed by the people here are far better for "real world" scenarios.
- The CSS code I created is based on the first thoughts that came to my mind and my own personal approach to the problem.
- My solution only works on symmetrical characters, like X, A, O, M. **It does not work on asymmetric characters like B, C, F, K or lower case letters.
- ** HOWEVER, this approach creates very interesting 'shapes' with asymmetric characters. Try changing the X to a K or to a lower case letter like an *h* or a *p* in the CSS :)

HTML

```
<span class="half-letter"></span>
```

SCSS

```
.half-character {
  display: inline-block;
  font: bold 350px/.8 Arial;
  position: relative;

  &:before, &:after {
    content: 'X'; //Change character here
    display: inline-block;
    width: 50%;
    overflow: hidden;
    color: #7db9e8;
  }
  &:after {
    position: absolute;
    top: 0;
    left: 50%;
    color: #1e5799;
    transform: rotateY(-180deg);
  }
}
```

edited Jul 31 '14 at 19:19

answered May 16 '14 at 20:21



[Ricardo Zea](#)

2,532 ● 6 ● 40 ● 60

You can also do it using SVG, if you wish:

```
var title = document.querySelector('h1'),
    text = title.innerHTML,
    svgTemplate = document.querySelector('svg'),
    charStyle = svgTemplate.querySelector('#text');

svgTemplate.style.display = 'block';
```

```
var space = 0;

for (var i = 0; i < text.length; i++) {
  var x = charStyle.cloneNode();
  x.textContent = text[i];
  svgTemplate.appendChild(x);
  x.setAttribute('x', space);
  space += x.clientWidth || 15;
}

title.innerHTML = '';
title.appendChild(svgTemplate);

<svg style="display: none; height: 100px; width: 100%" xmlns="http://www.w3.org/2000/svg"
xmlns:svg="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
version="1.1">
  <defs id="FooDefs">
    <linearGradient id="MyGradient" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="50%" stop-color="blue" />
      <stop offset="50%" stop-color="red" />
    </linearGradient>
  </defs>
  <text y="50%" id="text" style="font-size: 72px; fill: url(#MyGradient)"></text>
</svg>

<h1>This is not a solution X</h1>
```

<http://codepen.io/nicbell/pen/jGcbq>

edited Feb 6 at 10:43



Salman A

103k ● 36 ● 227 ● 307

answered May 15 '14 at 13:26



Nic Bell

183 ● 5

protected by **Josh Crozier** May 13 '14 at 16:05

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.

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