Web Style Sheets
CSS tips & tricks

See also the index of all tips.

CENTERING THINGS

A common task for CSS is to center text or images. In fact, there are three kinds of centering:

- Centering lines of text
- Centering a block of text or an image
- Centering a block or an image vertically

In recent implementations of CSS you can also use features from level 3, which allows centering absolutely positioned elements:

- Centering vertically in level 3
- Centering vertically and horizontally in level 3
- Centering in the viewport in level 3

CENTERING LINES OF TEXT

The most common and (therefore) easiest type of centering is that of lines of text in a paragraph or in a heading. CSS has the property 'text-align' for that:

```
P { text-align: center }
H2 { text-align: center }
```

renders each line in a P or in a H2 centered between its margins, like this:

The lines in this paragraph are all centered between the paragraph's margins, thanks to the value 'center' of the CSS property 'text-align'.

CENTERING A BLOCK OR IMAGE

Sometimes it is not the text that needs to be centered, but the block as a whole. Or, phrased differently: we want the left and right margin to be equal. The way to do that is to set the margins to 'auto'. This is normally used with a block of fixed width, because if the block itself is flexible, it will simply take up all the available width. Here is an example:

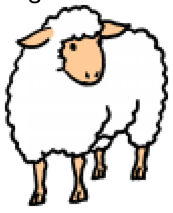
```
P.blocktext {
    margin-left: auto;
    margin-right: auto;
    width: 8em
}
...
<P class="blocktext">This rather...
```

This rather narrow block of text is centered. Note that the lines inside the block are not centered (they are left-aligned), unlike in the earlier example.

This is also the way to center an image: make it into block of its own and apply the margin properties to it. For example:

```
IMG.displayed {
    display: block;
    margin-left: auto;
    margin-right: auto }
...
<IMG class="displayed" src="..." alt="...">
```

The following image is centered:



CENTERING VERTICALLY

CSS level 2 doesn't have a property for centering things vertically. There will probably be one in CSS level 3 (see below). But even in CSS2 you can center blocks vertically, by combining a few properties. The trick is to specify that the outer block is to be formatted as a table cell, because the contents of a table cell *can* be centered vertically.

The example below centers a paragraph inside a block that has a certain given height. A separate example shows a paragraph that is centered vertically in the browser window, because it is inside a block that is absolutely positioned and as tall as the window.

```
DIV.container {
    min-height: 10em;
    display: table-cell;
    vertical-align: middle }
...
<DIV class="container">
  <P>This small paragraph...
</DIV>
```

This small paragraph is vertically centered.

CENTERING VERTICALLY IN CSS LEVEL 3

CSS level 3 offers other possibilities. At this time (2014), a good way to center blocks vertically without using absolute positioning (which may cause overlapping text) is still under discussion. But if you know that overlapping text will not be a problem in your document, you can use the 'transform' property to center an absolutely positioned element. For example:

This paragraph is vertically centered.

For a document that looks like this:

```
<div class=container3>
  <p>This paragraph…
</div>
```

the style sheet looks like this:

```
div.container3 {
   height: 10em;
   position: relative }              /* 1 */
div.container3 p {
   margin: 0;
   position: absolute;               /* 2 */
   top: 50%;                         /* 3 */
   transform: translate(0, -50%) }   /* 4 */
```

The essential rules are:

1. Make the container *relatively positioned*, which declares it to be a container for absolutely positioned elements.
2. Make the element itself *absolutely positioned*.
3. Place it halfway down the container with 'top: 50%'. (Note that 50%' here means 50% of the height of the container.)
4. Use a translation to move the element up by half its own height. (The '50%' in 'translate(0, -50%)' refers to the height of the element itself.)

Recently (since about 2015), another technique has also become available in several CSS implementations. It is based on the new 'flex' keyword for the 'display' property. This keyword is meant for use in graphical user interfaces (GUIs), but nothing stops you from using it in a document, if the document happens to have the right structure.

This paragraph is vertically centered.

the style sheet looks like this:

```
div.container5 {
  height: 10em;
  display: flex;
  align-items: center }
div.container5 p {
  margin: 0 }
```

CENTERING VERTICALLY AND HORIZONTALLY IN CSS LEVEL 3

We can extend both methods to center horizontally *and* vertically at the same time.

A side-effect of making the paragraph absolutely positioned is that it is then only as wide as it needs to be (unless we give it an explicit width, of course). In the example below, that's precisely what we want: We center a paragraph with just one word ("Centered!"), so the width of the paragraph should be exactly the width of that word.

Centered!

The yellow background is there to show that the paragraph is indeed only as wide as its contents. We assume the same mark-up as before:

```
<div class=container4>
  <p>Centered!
</div>
```

The style sheet is similar to the previous example with respect to the vertical centering. But we now move the element halfway across the container as well, with 'left:

50%', and at the same time move it leftwards by half its own width in the 'translate' transformation:

```
div.container4 {
    height: 10em;
    position: relative }
div.container4 p {
    margin: 0;
    background: yellow;
    position: absolute;
    top: 50%;
    left: 50%;
    margin-right: -50%;
    transform: translate(-50%, -50%) }
```

The next example below explains why the 'margin-right: -50%' is needed.

When the CSS formatter supports 'flex', it's even easier:

Centered!

with this style sheet:

```
div.container6 {
  height: 10em;
  display: flex;
  align-items: center;
  justify-content: center }
div.container6 p {
  margin: 0 }
```

i.e., the only addition is the 'justify-content: center'. Just like 'align-items' determines the vertical alignment of the container's contents, 'justify-content' determines the horizontal alignment. (It's actually a bit more complex, as their names suggest, but in a simple case that's how it works.) A side-effect of 'flex' is that the child element, the P in this case, is automatically made as small as possible.

CENTERING IN THE VIEWPORT IN CSS LEVEL 3

The default container for absolutely positioned elements is the viewport. (In case of a browser, that means the browser window). So centering an element in the viewport is very simple. Here is a complete example. (This example uses HTML5 syntax.)

```
<html>
  <style>
    body {
        background: white }
    section {
        background: black;
        color: white;
        border-radius: 1em;
        padding: 1em;
        position: absolute;
        top: 50%;
        left: 50%;
        margin-right: -50%;
        transform: translate(-50%, -50%) }
  </style>
  <section>
    <h1>Nicely centered</h1>
    <p>This text block is vertically centered.
    <p>Horizontally, too, if the window is wid
  </section>
```

You can see the result in a separate document.

The 'margin-right: -50%' is needed to compensate the 'left: 50%'. The 'left' rule reduces the available width for the element by 50%. The renderer will thus try to make lines that are no longer than half the width of the container. By saying that the right margin of the element is further to the right by that same amount, the maximum line length is again the same as the container's width.

Try resizing the window: You'll see that each sentence is on one line when the window is wide enough. Only when the window is too narrow for the whole sentence will the sentence be broken over several lines. When you remove the 'margin-right: -50%' and resize the window again, you'll see that the sentences will be broken already when the window is still twice as wide as the text lines.

(Using 'translate' for centering in the viewport was first proposed by “Charlie” in an answer on Stack Overflow.)

Bert Bos, style activity lead

Created 5 May 2001;
Last updated Wed 26 Sep 2018 04:41:20 AM UTC

LANGUAGES

- Azərbaycan
- Български
- Deutsch
- Ελληνικά
- English
- Español
- Français
- Bahasa Indonesia
- Norsk
- Nederlands
- Polski
- Português
- Português brasileiro
- Русский
- ไทย
- Tagalog
- Українська
- Tiếng Việt
- 简体中文
- 繁體中文

About the translations