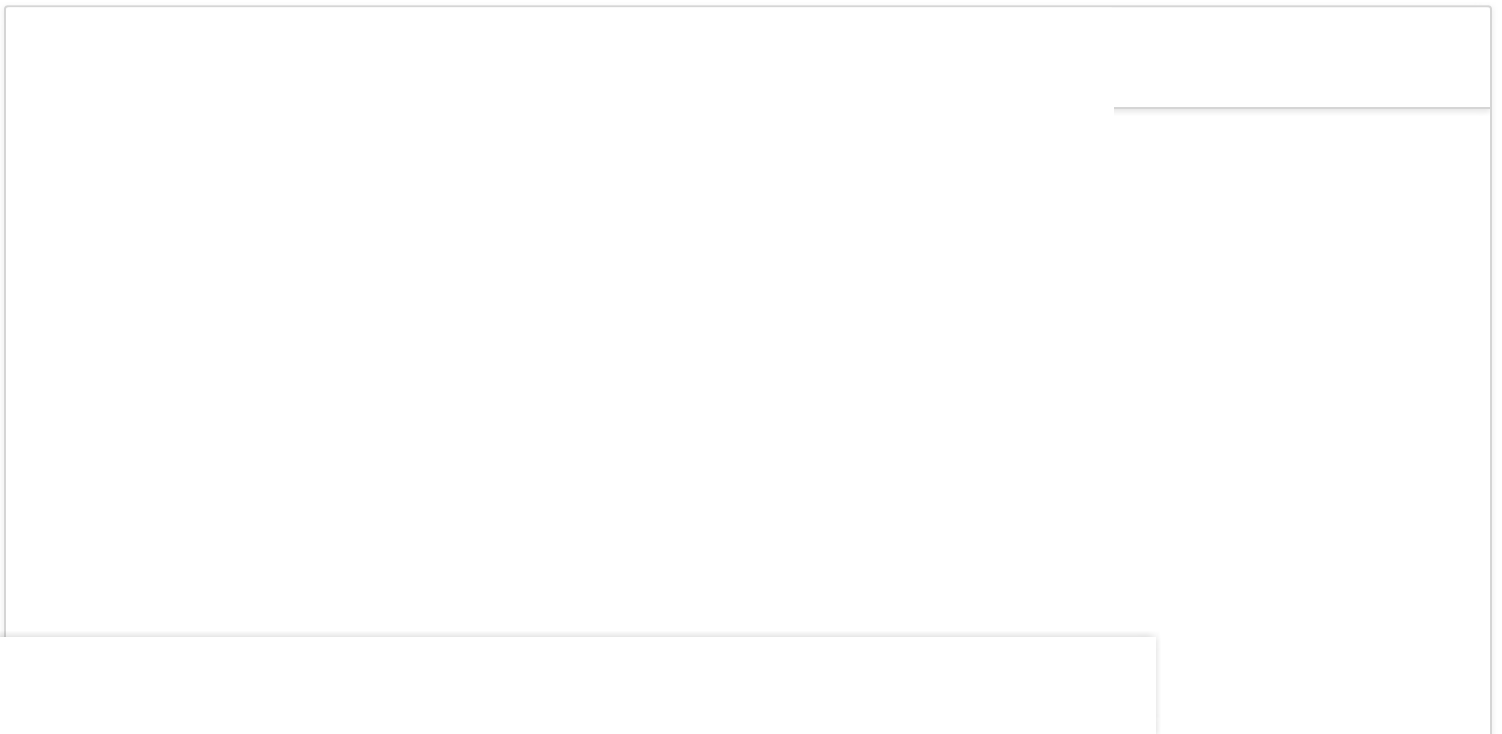


Quick Tip: The Simplest Way To Center Elements Vertically And Horizontally

September 2nd 2015 **CSS** | **Tip** | **Flexbox**

Flexbox is a relatively new addition to the CSS world and we keep finding more and more excellent applications for it. We can even solve the age-old problem with [centering an element both vertically and horizontally](#) with CSS. It is easier than you can imagine - it is only three lines of code, doesn't require you to specify the size of the centered element and is responsive-friendly!

This technique works in all modern browsers - IE10+, Chrome, Firefox and Safari (with the `-webkit-` prefix). See the full compatibility table [here](#).



Edit

Run

The HTML

The idea, of course, revolves around flexbox. First, we create a container in which we want everything to be centered:

```
<div class="container">
  <!--// Any content in here will be centered.-->
  
</div>
```

You can place this container div anywhere you want. In the live example above we've made it take up the whole width and height of the page.

The CSS

As we said earlier, we will be using only three lines of code. Here they are:

```
.container{
  display: flex;
  justify-content: center;
  align-items: center;
}
```

Every flex container has two axis for positioning elements. The main axis is declared with the `flex-direction` property (can be `row` or `column`, see [docs](#)). By omitting this rule, we've left the flex direction to its default `row` value.

Now all that we need to do is center both axis. It couldn't get any simpler:

1. Make the display type `flex`, so that we activate flexbox mode.
2. `justify-content` defines where flex items will align according to the main axis (horizontally in our case).
3. `align-items` does the same with the axis perpendicular to the main one (vertically in our case).

Now that we have set the rules for the vertical and the horizontal alignment to *center*, any element we add its dimensions

Conclusion

There are lots of other techniques for centering content with CSS, but flexbox makes this a whole lot simpler and more elegant. If you wish to learn more about it, check out these resources:

- A complete guide to flexbox - [here](#).
- MDN: Using CSS flexible boxes (a long read) - [here](#).
- Flexbox in 5 minutes - [here](#).

Related Articles

Quick Tip: The Best Way To Make Sticky Footers

[CSS](#) | [Flexbox](#)

MicroTut: Centering a Div Both Horizontally And Vertically

[CSS](#) | [jQuery](#)

Quick Tip: The Easiest Way To Make Responsive Headers

CSS | [Quick Learn](#)

Comments 21

[Log In To Comment](#)



Joe Twumasi

2 years ago

What about IE compatibility and other older browsers?

Nice article on centering elements vertically and horizontally.



Danny Markov

2 years ago

Thanks! IE 10+ support flexbox, as well as all other new(ish) browsers. See [here](#).



Natalie

2 years ago

Well, that looks simple enough! Thanks for sharing :D



Maurizio

2 years ago

I think flex-box is very useful, but we have to keep in mind that is not fully supported by browsers (ie. IE 8 - 9)

p.s. sorry for my awful english



Danny Markov

Thanks Maurizio, you're right. If your visitors use old version of IE your will have to stick with the usual techniques for centering content.



MaxArt

2 years ago

IE < 10 doesn't just "not fully support" flexbox, it doesn't support it at all!
But it's also time to ignore that old browser, honestly.



Alexander Dimitrov

2 years ago

Yeah i wish :(I am WP Dev, and WordPress officially does support ie9, so your site must be ie0 compatible, the only reason i am not using it yet. But true, we should move on finally...



Jayesh Morker

2 years ago

How to add prefix for "Safari" browser?



Danny Markov

2 years ago

Thanks for bringing this to our attention, Jayesh. We didn't mention it in the article but for this technique to work in Safari prior to version 9 you need to prefix the flexbox properties with `-webkit-`. This also applies to iOS Safari.



Gindeev

2 years ago

Doesn't work on iPhone iOS 7.0.4 or iPad iOS 7.1.4 bro!



Danny Markov

2 years ago

It does work, you only need to prefix it.



Gindeev

2 years ago

Oh yes! It works!!!



Amir Off

2 years ago

Thanks Danny, I was struggling until I found this article. Traditional centering techniques didn't help in my case!



2 years ago

well done!



sunpietro

2 years ago

There's also another way to center elements inside:

```
.container {  
  display: flex;  
}  
  
.container img {  
  margin: auto;  
}
```



SelenIT

2 years ago

Simple flexbox-based centering solutions may lead to some surprise when the content becomes taller than the container. Just adding `overflow: auto` to container won't help: <http://jsfiddle.net/wp8av11c/1/>



Tadas

2 years ago

This looks like a better solution for me:

```
position: absolute;  
top: 50%; left: 50%;  
transform: translate(-50%, -50%);
```



DidouPh

2 years ago

Indeed, you're right !

So the best option is to >

1 - set default to :

```
position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);
```

2 - if the browser is supported, use :

```
display: flex; justify-content: center; align-items: center;
```



Divy Singh Rathore

2 years ago



Md. Elias

2 years ago

That is awesome... waiting for next awesome article :)



Abbas Uddin

7 months ago

How about browser hack?

```
div.mybox{  
padding: 45% 0%;  
margin-left: auto;  
margin-right: auto;  
}
```

Copyright © 2017 Zine EOOD.

All Rights Reserved.