How to horizontally center a <div>?

Ask Question

3919

931

How can I horizontally center a `<div>` within another `<div>` using CSS?

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

html css alignment centering

edited Oct 30 '18 at 12:21

community wiki
32 revs, 23 users 13%
Mosh Feu

4 Of those great answers, I just want to highlight that you must give "#inner" a "width", or it will be "100%", and you can't tell if it's already centered. – Jony Nov 7 '17 at 8:22

Home

98 Answers

1 2 3 4 next

4360

✓

You can apply this CSS to the inner `<div>` :

```
#inner {
  width: 50%;
  margin: 0 auto;
}
```

Of course, you don't have to set the `width` to `50%` . Any width less than the containing `<div>` will work. The `margin: 0 auto` is what does the actual centering.

If you are targeting IE8+, it might be better to have this instead:

```
#inner {
  display: table;
  margin: 0 auto;
}
```

It will make the inner element center horizontally and it works without setting a specific `width` .

Working example here:

```
#inner {
  display: table;
  margin: 0 auto;
}


<div id="outer" style="width:100%">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet Expand snippet

edited Feb 28 '17 at 13:44

community wiki
11 revs, 7 users 31%
bharadhwaj

10 For the vertical centering I usually use "line-height" (line-height == height). This is simple and nice but it's only working with a one line content text :) – Nicolas Guillaume Jun 23 '10 at 12:36

90 You have to use the !DOCTYPE tag on your html page to make it work well on IE. – Fabio Jan 28 '12 at 14:23

14 Note that it may be necessary to add "float:none;" for the #inner. – Mert Mertce Sep 27 '13 at 8:30

14 You also set the top and bottom margins to 0, which is unrelated. Better putting `margin-left: auto; margin-right: auto` I think. – Emmanuel Touzery Feb 8 '14 at 22:45

12 Not necessarily `margin:0 auto` : it can be `margin: <whatever_vertical_margin_you_need> auto` second being the horizontal margin. – YakovL May 3 '16 at 19:07

This worked for me:

0

```
#outer {
     width: 100%;
     text-align: center;
   }

   #inner {
     display: inline-block;
   }
   <div id="outer">
       <div id="inner">Foo foo</div>
   </div>
```

edited 2 days ago

community wiki
2 revs, 2 users 96%
shikou lardji

1

```
#outer{
    width: 100%;

    /* Firefox */
    display: -moz-box;
    -moz-box-pack: center;
    -moz-box-align: center;

    /* Safari and Chrome */
    display: -webkit-box;
    -webkit-box-pack: center;
    -webkit-box-align: center;

    /* W3C */
    display: box;
    box-pack: center;
    box-align: center;
}
#inner{
    width: 50%;
}
```

answered 2 days ago

community wiki
mohamed laribi

2

you can just simply use **flexbox** like this:

```
#outer {
  display: flex;
  justify-content: center
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet Expand snippet

Apply Autoprefixer for all browser support:

```
#outer {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  width: 100%;
  -webkit-box-pack: center;
      -ms-flex-pack: center;
          justify-content: center
}
```

OR ELSE

use **transform**:

```
#inner {
      position: absolute;
      left: 50%;
      transform: translate(-50%)
    }
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

with Autoprefixer:

```
#inner {
  position: absolute;
  left: 50%;
  -webkit-transform: translate(-50%);
      -ms-transform: translate(-50%);
          transform: translate(-50%)
}
```

answered Apr 4 at 11:59

community wiki
Shashin Bhayani

2

Make it simple!

```
#outer {
  display: flex;
  justify-content: center;
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet Expand snippet

answered Mar 26 at 15:27

community wiki
Hassan Siddiqui

2

```
.outer
{
  background-color: rgb(230,230,255);
  width 100%;
  height: 50px;
}
.inner
{
  background-color: rgb(200,200,255);
  width: 50%;
  height: 50px;
  margin: 0 auto;
}
```

```
<div class="outer">
  <div class="inner">
    margin 0 auto
  </div>
</div>
```

Run code snippet   Expand snippet

answered Mar 7 at 14:10

community wiki
pavelbere

8

As of 2019 since this question still gets a lot of hits a very simple and cross browser answer to horizontal center is to apply this rule to the parent element:

```
.parentBox {
 display: flex;
 justify-content: center
}
```

answered Feb 12 at 2:47

community wiki
Ady Ngom

1

```
div{
   width:100px;
   height:100px;
   margin:0 auto;
  }
```

for normal thing if you using div static way

if you want div to be in centered when div is absolute to its parent here is example:

```
.parentdiv{
   position:relative;
   height:500px;
}

.child_div{
   position:absolute;
   height:200px;
   width:500px;
   left:0;
   right:0;
   margin:0 auto;
}
```

edited Feb 7 at 14:18

community wiki
2 revs, 2 users 89%
Mohammed Rabiulla RABI

I found similar with `margin-left` , but can be `left` as well.

1

```
#inner{
    width: 100%;
    max-width: 65px; /*to adapt to screen width. Can be whatever you
    left: 65px; /*this has to be approximately the same as the max-w
}
```

answered Dec 29 '18 at 21:02

community wiki
iorgu

3

Here is another way to center horizontally using **flexbox** and without specifying any width to inner container. The idea is to use pseudo elements that will push the inner content from the right and the left.

Using `flex:1` on pseudo element will make them fill the remaining spaces and take equal size and the inner container will get centered.

```
.container {
  display: flex;
  border: 1px solid;
}

.container:before,
.container:after {
  content: "";
  flex: 1;
}

.inner {
  border: 1px solid red;
  padding: 5px;
}
```

```
<div class="container">
  <div class="inner">
    Foo content
  </div>
</div>
```

Run code snippet   Expand snippet

We can also consider the same situation for vertical alignment by simply changing the direction of flex to column:

```
.container {
  display: flex;
  flex-direction: column;
  border: 1px solid;
  min-height: 200px;
}

.container:before,
.container:after {
  content: "";
  flex: 1;
}

.inner {
  border: 1px solid red;
  padding: 5px;
}
```

```
<div class="container">
  <div class="inner">
    Foo content
  </div>
</div>
```

Run code snippet   Expand snippet

edited Dec 14 '18 at 21:26

community wiki
3 revs, 2 users 95%
Temani Afif

1

this worked for me:

```
   #inner {
    position: absolute;
    margin: 0 auto;
    left: 0;
    width: 7%;
    right: 0;
}
```

In this code, you to be determine width of element.

answered Nov 25 '18 at 8:40

community wiki
Fatemeh Khosravi Farsani

3

You can do it buy using flex-box witch is a good technique these days. For using flex-box you should give `display: flex;` and `align-items: center;` to your parent or `#outer` div element. The code should be like this :

```
#outer {
  display: flex;
  align-items: center;
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

This should center your child or `#inner` div horizontally. But you can't actually see any changes. Because our `#outer` div has no height or in other words, its height is set to auto so it has the same height as all of its child elements. So after a little of visual styling, the result code should be like this :

```
#outer {
  height: 500px;
  display: flex;
  align-items: center;
  background-color: blue;
}

#inner {
  height: 100px;
  background: yellow;
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

You can see `#inner` div is now centered. Flex-box is the new method of positioning elements in horizontal or vertical stacks with CSS and it's got 96% of global browsers compatibility. So you are free to use it and if you want to find out more about Flex-box visit CSS-Tricks article, that is the best place to learn using Flex-box in my opinion.

answered Sep 10 '18 at 23:06

community wiki
Morteza Sadri

3

This will surely center your `#inner` both horizontally and vertically. This is also compatible in all browsers. I just added extra styling just to show how it is centered.

```
#outer {
  background: black;
  position: relative;
  width:150px;
  height:150px;
}

#inner {
  background:white;
  position: absolute;
  left:50%;
  top: 50%;
  transform: translate(-50%,-50%);
  -webkit-transform: translate(-50%,-50%);
  -moz-transform: translate(-50%,-50%);
  -o-transform: translate(-50%,-50%);
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

But of course if you only want it horizontally aligned, This may help you.

```
#outer {
  background: black;
  position: relative;
  width:150px;
  height:150px;
}

#inner {
  background:white;
  position: absolute;
  left:50%;
  transform: translate(-50%,0);
  -webkit-transform: translate(-50%,0);
  -moz-transform: translate(-50%,0);
  -o-transform: translate(-50%,0);
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

edited Sep 7 '18 at 22:02

community wiki

2 revs
Mark Salvania

2

One of the easiest ways you can do it is by using `display: flex`. The outer div just needs to have display flex, and the inner needs `margin: 0 auto` to make it centered horizontally.

To center vertically and just center a div within another div, please look at the comments of the .inner class below

```css
.wrapper {
  display: flex;
  /* Adding whatever height & width we want */
  height: 300px;
  width: 300px;
  /* Just so you can see it is centered */
  background: peachpuff;
}

.inner {
  /* center horizontally */
  margin: 0 auto;
  /* center vertically */
  /* margin: auto 0; */
  /* center */
  /* margin: 0 auto; */
}
```

```html
<div class="wrapper">
  <div class="inner">
    I am horizontally!
  </div>
</div>
```

Run code snippet   Expand snippet

answered Sep 7 '18 at 21:56

community wiki
drewkiimon

21

You can use `display: flex` for your outer div and to horizontally center you have to add `justify-content: center`

```
#outer{
    display: flex;
    justify-content: center;
}
```

or you can visit [w3schools - CSS flex Property]() for more ideas.

edited Aug 20 '18 at 12:57

community wiki
2 revs, 2 users 75%
Milan Panigrahi

1

```
#outer {
  width: 160px;
  padding: 5px;
  border-style: solid;
  border-width: thin;
  display: block;
}

#inner {
  margin: auto;
  background-color: lightblue;
  border-style: solid;
  border-width: thin;
  width: 80px;
  padding: 10px;
  text-align: center;
}
```

```
<div id="outer">
  <div id="inner">Foo foo</div>
</div>
```

Run code snippet   Expand snippet

edited Aug 14 '18 at 14:58

community wiki
2 revs
Chris

3

The best I have used in my various projects is

```
<div class="outer">
    <div class="inner"></div>
</div>
.outer{
  width: 500px;
  height: 500px;
  position: relative;
  background: yellow;
}
.inner{
  width: 100px;
  height: 100px;
  background:red;
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
}
```

fiddle link

answered Aug 7 '18 at 10:01

community wiki
Jaison

-2

```
<!DOCTYPE html>
<html>
  <head>
    <title>Center</title>
    <style>
    .outer{
          text-align: center;
           }
    .inner{
          width: 500px;
          margin: 0 auto;
          background: brown;
          color: red;
    }

</style>

  </head>
  <body>

    <div class="outer">
      <div class="inner">This DIV is centered</div>
    </div>

  </body>
</html>
```

Run code snippet Expand snippet

plz try this will work without html center tag

answered Jul 15 '18 at 15:00

community wiki
MANGESH SUPE

19

If anyone would like a jQuery solution for center align these divs:

```
$(window).bind("load", function() {
    var wwidth = $("#outer").width();
    var width = $('#inner').width();
    $('#inner').attr("style", "padding-left: " + wwidth / 2 + "px; m
width / 2 + "px;");
});
```

edited Mar 19 '18 at 9:34

community wiki

6 revs, 4 users 79%
sarath

24

If width of the content is unknown you can use the following method. Suppose we have these two elements:

- `.outer` -- full width
- `.inner` -- no width set (but a max-width could be specified)

Suppose the computed width of the elements are 1000px and 300px respectively. Proceed as follows:

1. Wrap `.inner` inside `.center-helper`
2. Make `.center-helper` an inline block; it becomes the same size as `.inner` making it 300px wide.
3. Push `.center-helper` 50% right relative to its parent; this places its left at 500px wrt. outer.
4. Push `.inner` 50% left relative to its parent; this places its left at -150px wrt. center helper which means its left is at 500 - 150 = 350px wrt. outer.
5. Set overflow on `.outer` to hidden to prevent horizontal scrollbar.

Demo:

```
body {
  font: medium sans-serif;
}

.outer {
  overflow: hidden;
  background-color: papayawhip;
}

.center-helper {
  display: inline-block;
  position: relative;
  left: 50%;
  background-color: burlywood;
}

.inner {
  display: inline-block;
  position: relative;
  left: -50%;
  background-color: wheat;
}
```

```
<div class="outer">
  <div class="center-helper">
    <div class="inner">
      <h1>A div with no defined width</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.<b
          Duis condimentum sem non turpis consectetur blandit.<br>
          Donec dictum risus id orci ornare tempor.<br>
          Proin pharetra augue a lorem elementum molestie.<br>
          Nunc nec justo sit amet nisi tempor viverra sit amet a ip
    </div>
  </div>
</div>
```

Run code snippet   Expand snippet

```
.outer {
    overflow: hidden;
}
.center-helper {
    float: left;
    position: relative;
    left: 50%;
}
.inner {
    float: left;
    position: relative;
    left: -50%;
}
```

edited Mar 17 '18 at 21:12

community wiki
3 revs, 2 users 92%
Salman A

179

Some posters have mentioned the CSS 3 way to center using `display:box` .

This syntax is outdated and shouldn't be used anymore. [See also this post].

So just for completeness here is the latest way to center in CSS 3 using the **Flexible Box Layout Module**.

So if you have simple markup like:

```
<div class="box">
  <div class="item1">A</div>
  <div class="item2">B</div>
  <div class="item3">C</div>
</div>
```

...and you want to center your items within the box, here's what you need on the parent element (.box):

```
.box {
    display: flex;
    flex-wrap: wrap; /* Optional. only if you want the items to wrap
    justify-content: center; /* For horizontal alignment */
    align-items: center; /* For vertical alignment */
}
```

Show code snippet

If you need to support older browsers which use older syntax for flexbox here's a good place to look.

edited Mar 17 '18 at 20:14

community wiki
8 revs, 4 users 86%
Danield

4 The Flexbox specification has gone through 3 major revisions. The most recent draft is from Sept 2012, which officially deprecates all previous drafts. However, browser support is spotty (particularly old Android browsers): stackoverflow.com/questions/15662578/… – cimmanon Oct 1 '13 at 20:33

3 @WouterVanherck it depends on the `flex-direction` value. If it is 'row' (the default) - then `justify-content: center;` is for the horizontal alignment (like I mentioned in the answer) If it is 'column' - then `justify-content: center;` is for the vertical alignment. – Danield Mar 22 '17 at 12:29

-8

Centering: Auto-width Margins

This box is horizontally centered by setting its right and left margin widths to "auto". This is the preferred way to accomplish horizontal centering with CSS and works very well in most browsers with CSS 2 support. Unfortunately, Internet Explorer 5/Windows does not respond to this method - a shortcoming of that browser, not the technique.

There is a simple workaround. (A pause while you fight back the nausea induced by that word.) Ready? Internet Explorer 5/Windows incorrectly applies the CSS "text-align" attribute to block-level elements. Declaring "text-align:center" for the containing block-level element (often the BODY element) horizontally centers the box in Internet Explorer 5/Windows.

There is a side effect of this workaround: the CSS "text-align" attribute is inherited, centering inline content. It is often necessary to explicitly set the "text-align" attribute for the centered box, counteracting the effects of the Internet Explorer 5/Windows workaround. The relevant CSS follows.

```
body {
    margin: 50px 0px;
    padding: 0px;
    text-align: center;
}

#Content {
    width: 500px;
    margin: 0px auto;
    text-align: left;
    padding: 15px;
    border: 1px dashed #333;
    background-color: #EEE;
}
```

http://bluerobot.com/web/css/center1.html

edited Mar 17 '18 at 20:10

community wiki
2 revs, 2 users 73%
Ray

5 That's not centering the div, that's centering the text. – clickbait Jun 12 '15 at 23:50

335

The best approaches are with CSS 3.

Box model:

```
#outer{
    width: 100%;

    /* Firefox */
    display: -moz-box;
    -moz-box-pack: center;
    -moz-box-align: center;

    /* Safari and Chrome */
    display: -webkit-box;
    -webkit-box-pack: center;
    -webkit-box-align: center;

    /* W3C */
    display: box;
    box-pack: center;
    box-align: center;
}
#inner{
    width: 50%;
}
```

According to your usability you may also use the `box-orient, box-flex, box-direction` properties.

Flex:

```
#outer {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: center;
    align-items: center;
}
```

Read more about centering the child elements

- Link 2

- Link 3
- Link 4

And **this explains why the box model is the best approach**:

- *Why is the W3C box model considered better?*

edited Mar 17 '18 at 20:07

community wiki
12 revs, 7 users 58%
Konga Raju

22 Make sure you read this answer first before you go about implementing this solution. – cimmanon Apr 24 '13 at 18:51

4 Safari, as of now, still requires `-webkit` flags for flexbox (`display: -webkit-flex;` and `-webkit-align-items: center;` and `-webkit-justify-content: center;`) – Joseph Hansen Jul 23 '15 at 15:59

49

For Firefox and Chrome:

```
<div style="width:100%;">
  <div style="width: 50%; margin: 0px auto;">Text</div>
</div>
```

Run code snippet Expand snippet

For Internet Explorer, Firefox, and Chrome:

```
<div style="width:100%; text-align:center;">
  <div style="width: 50%; margin: 0px auto; text-align:left;">Text<
</div>
```

Run code snippet   Expand snippet

The `text-align:` property is optional for modern browsers, but it is necessary in Internet Explorer Quirks Mode for legacy browsers support.

edited Mar 17 '18 at 19:13

community wiki
7 revs, 7 users 47%
ch2o

5 There is no need for text-align property. It's completely un-necessary. – Touhid Rahman May 23 '13 at 5:29

80

Set the `width` and set `margin-left` and `margin-right` to `auto` . That's **for horizontal only**, though. If you want both ways, you'd just do it both ways. Don't be afraid to experiment; it's not like you'll break anything.

edited Mar 17 '18 at 19:11

community wiki
4 revs, 4 users 33%
Sneakyness

2

Yes, this is short and clean code for horizontal align. I hope you like this code.

```
.classname {
   display: box;
   margin: 0 auto;
   width: 500px /* Width set as per your requirement. */;
}
```

edited Mar 17 '18 at 19:10

community wiki
3 revs, 3 users 59%
BANTIMATHUR

1

Try out this:

```
#outer{
    display: inline-block;
    height: 100%;
    vertical-align: middle;
}

#outer > #inner{
    display: inline-block;
    font-size: 19px;
    margin: 20px;
    max-width: 320px;
    min-height: 20px;
    min-width: 30px;
    padding: 14px;
    vertical-align: middle;
}
```

edited Mar 17 '18 at 19:08

community wiki
2 revs, 2 users 72%
Ajay Gupta

40

Another solution for this without having to set a width for one of the elements is using the CSS 3 `transform` attribute.

```
#outer {
  position: relative;
}

#inner {
  position: absolute;
  left: 50%;

  transform: translateX(-50%);
}
```

The trick is that `translateX(-50%)` sets the `#inner` element 50 percent to the left of its own width. You can use the same trick for vertical alignment.

Here's a **Fiddle** showing horizontal and vertical alignment.

More information is on Mozilla Developer Network.

edited Mar 17 '18 at 19:08

community wiki

2 revs, 2 users 89%
Kilian Stinson

2 One may need vendor prefixes as well : `-webkit-transform: translate(-50%,0); -moz-transform: translate(-50%,0); -ms-transform: translate(-50%,0); -khtml-transform: translate(-50%,0); -o-transform: translate(-50%,0);` – Skippy le Grand Gourou Sep 2 '15 at 13:48

1

Instead of multiple wrappers and/or auto margins, this simple solution works for me:

```
<div style="top:50%; left:50%;
    height:100px; width:100px;
    margin-top:-50px; margin-left:-50px;
    background:url('lib/loading.gif') no-repeat center #fff;
    text-align:center;
    position:fixed; z-index:9002;">Loading...</div>
```

It puts the div at the center of the view (vertical and horizontal), sizes and adjusts for size, centers background image (vertical and horizontal), centers text (horizontal), and keeps div in the view and on top of the content. Simply place in the HTML `body` and enjoy.

edited Mar 17 '18 at 19:07

community wiki
2 revs, 2 users 90%
Cees Timmerman

1

First of all: You need to give a width to the second div:

For example:

HTML

```
<div id="outter">
    <div id="inner"Centered content">
    </div
</div>
```

CSS:

```
 #inner{
     width: 50%;
     margin: auto;
}
```

Note that if you don't give it a width, it will take the whole width of the line.

edited Mar 17 '18 at 18:56

community wiki
2 revs, 2 users 74%
user3968801

1 2 3 4 next