How to horizontally center a <div>?

Asked 10 years, 11 months ago Active 20 days ago Viewed 3.8m times

4056

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

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

961

html css alignment centering

edited Oct 30 '18 at 12:21

community wiki
32 revs, 23 users 13%
Mosh Feu

5 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

101 Answers

1 2 3 4 next

4496

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

```
#inner {
```

✓

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;
  border: 1px solid black;
}

#outer {
  border: 1px solid red;
  width:100%
}


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

Run code snippet   Expand snippet

edited Jun 6 at 18:45

community wiki
12 revs, 8 users 28%
bharadhwaj

91 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

1195

If you don't want to set a fixed width on the inner `div` you could do something like this:

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

#inner {
  display: inline-block;
}
```

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

Run code snippet Expand snippet

That makes the inner `div` into an inline element that can be centered with `text-align` .

edited Nov 22 '17 at 11:34

community wiki
5 revs, 4 users 62%
Alfred

12 @SabaAhang the correct syntax for that would be `float: none;` and is probably only needed because #inner has inherited a `float` of either

value. – pmoleri Nov 18 '16 at 21:52

357

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%;
}
```

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

Run code snippet Expand snippet

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

Flex:

```
    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 May 15 at 13:38

community wiki
13 revs, 9 users 50%
Konga Raju

it also works for me when inner div have float: left; – Tareq Nov 12 '12 at 6:30

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

237

Suppose that your div is `200px` wide:

```
.centered {
  position: absolute;
  left: 50%;
  margin-left: -100px;
}
```

Make sure the parent element is positioned i.e. relative, fixed, absolute, or sticky.

https://jsfiddle.net/gjvfxxdj/

edited Sep 25 '17 at 10:48

community wiki
7 revs, 7 users 48%
nuno_cruz

This doesn't work in Safari – cesards Aug 8 '15 at 9:05

I don't like this solution because when the inner element is too broad for the screen, you can't scroll over the whole element horizontally. margin: 0 auto works better. – Aloso Dec 30 '15 at 4:02

why do u put margin left: -100, this will not work – Robert Limanto Nov 29 '16 at 23:51

I've read that it's the only method that will work in IE6/7 – Andy Oct 31 '17 at 7:25

margin-left: auto; margin-right: auto; centres a block level element – killscreenmike Nov 10 '17 at 19:15

220

I've created this example to show how to **vertically** and **horizontally** `align` .

Code is basically this:

```
#outer {
  position: relative;
}
```

and...

```
#inner {
  margin: auto;
  position: absolute;
  left:0;
  right: 0;
  top: 0;
  bottom: 0;
}
```

and it will stay in the `center` even when you **re-size** your screen

4 revs, 4 users 60%
Tom Maton

12 +1 for this method, I was about to answer with it. Note that you must declare a width on the element you wish to center horizontally (or height if centering vertically). Here's a comprehensive explanation: codepen.io/shshaw/full/gEiDt. One of the more versatile and widely-supported methods of centering elements vertically and/or horizontally. – stvnrynlds Dec 16 '13 at 18:27

6 You cannot use padding within the div, but if you want to give the illusion use a border of the same color. – Squirrl Jul 9 '14 at 11:45

I think for this method to work, you need to set the with and height of inner div – Nicolas S.Xu Nov 29 '15 at 21:39

190

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.

Danield

what do you mean by "syntax is outdated", is it deprecated? – Konga Raju Sep 6 '13 at 10:18

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

This worked for me in Chrome when Justin Poliey's version didn't. – Vern Jensen Jun 29 '16 at 2:50

Isn't the "justify-content: center;" for the vertical alignment and the "align-items: center;" for the horizontal alignment? – Wouter Vanherck Mar 22 '17 at 12:04

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

131

If you don't want to set a fixed width and don't want the extra margin, add `display: inline-block` to your element.

You can use:

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

edited Oct 29 '13 at 17:23

community wiki
Salman for Hire

3 same requirements as display:inline-block too (quirksmode.org/css/display.html) – montrealmike Sep 11 '12 at 15:09

I used this, too, but I've never encountered `display: table;` before. What does it do? – Matt Cremeens Jul 31 '17 at 11:25

85

Centering a div of unknown height and width

Horizontally and vertically. It works with reasonably modern browsers (Firefox, Safari/WebKit, Chrome, Internet Explorer 10, Opera, etc.)

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


<div class="content">This works with any content</div>
```

Run code snippet Expand snippet

Tinker with it further on Codepen or on JSBin.

edited Mar 17 '18 at 18:49

community wiki
6 revs, 4 users 81%
iamnotsam

This is best solution. – Mohammad Ayoub Khan Jul 3 at 1:12

84

It cannot be centered if you don't give it a width, otherwise it will take, by default the whole horizontal space.

edited May 9 '11 at 17:23

community wiki
gizmo

46 and if you don't know the width? Say because the content is dynamic? – gman Jun 2 '11 at 15:45

max-width? what about that? – Will Hoskings Mar 17 '18 at 22:32

CSS3's box-align property

```
    display:box;
    box-orient:horizontal;

    box-pack:center;
    box-align:center;
}
```

edited Oct 29 '13 at 17:23

community wiki
neoneye

7 Make sure you read this answer first before you go about implementing this solution. – cimmanon Apr 24 '13 at 18:52

82

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

58

I recently had to center a "hidden" div (ie, display:none;) that had a tabled form within it that needed to be centered on the page. I wrote the following jQuery to display the hidden div & then update the CSS to the automatic generated width of the table and change the margin to center it. (The display toggle is triggered by clicking on a link, but this code wasn't neccessary to display.)

NOTE: I'm sharing this code because Google brought me to this Stack Overflow solution & everything would have worked except that hidden elements don't have any width & can't be resized/centered until after they are displayed.

```
$(function(){
  $('#inner').show().width($('#innerTable').width()).css('margin','0 auto');
});
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<div id="inner" style="display:none;">
```

```
      <tr><td>Email:</td><td><input type="text"></td></tr>
      <tr><td>Email:</td><td><input type="submit"></td></tr>
    </table>
  </form>
</div>
```

Run code snippet Expand snippet

edited Aug 4 '16 at 13:09

community wiki
2 revs, 2 users 65%
James Moberg

53

The way I usually do it is using absolute position:

```
#inner{
    left: 0;
    right: 0;
    margin-left: auto;
    margin-right: auto;
    position: absolute;
}
```

The outer div doesn't need any extra propertites for this to work.

edited May 31 '16 at 13:19

community wiki
3 revs, 2 users 95%
william44isme

This may not work if you have other divs below the centered div. – NoChance Jul 26 '18 at 7:59

50

For Firefox and Chrome:

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>
</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

text-align is actually necessary for it to work in IE quicks mode, so if you don't mind adding a little expression to support older browsers keep it there. (IE8 with IE8 rules and IE7 rules both work without text-align, so may be it's only IE6 and older that are concerned) – heytools Nov 4 '17 at 2:02

45

This is my answer.

```
#outerDiv {
  width: 500px;
}

#innerDiv {
  width: 200px;
```

```
<div id="outerDiv">
  <div id="innerDiv">Inner Content</div>
</div>
```

Run code snippet

Expand snippet

edited Feb 24 '17 at 8:37

community wiki
4 revs, 4 users 39%
Ankit Jain

▲

43

▼

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

`translate(-50%,0); -khtml-transform: translate(-50%,0); -o-transform: translate(-50%,0);` – Skippy le Grand Gourou Sep 2 '15 at 10:10

40

Chris Coyier who wrote an excellent post on 'Centering in the Unknown' on his blog. It's a roundup of multiple solutions. I posted one that isn't posted in this question. It has more browser support then the flexbox-solution, and you're not using `display: table;` which could break other things.

```
/* This parent can be any width and height */
.outer {
  text-align: center;
}

/* The ghost, nudged to maintain perfect centering */
.outer:before {
  content: '.';
  display: inline-block;
  height: 100%;
  vertical-align: middle;
  width:0;
  overflow:hidden;
}

/* The element to be centered, can
   also be of any width and height */
.inner {
  display: inline-block;
  vertical-align: middle;
  width: 300px;
}
```

edited Mar 10 '16 at 16:50

community wiki
3 revs, 3 users 93%
Willem de Wit

35

I realize I'm pretty late to the game, but this is a very popular question, and I recently found an approach I haven't seen mentioned anywhere here, so I figured I'd document it.

```
#outer {
```

```
#inner {

    position: relative;
    left: -50%;
}
```

EDIT: both elements must be the same width to function correctly.

edited Jul 19 '13 at 14:16

community wiki
BenjaminRH

Just set this rule for `#inner` only: `#inner { position:relative; left:50%; transform:translateX(-50%); }` . This works for any width. – Jose Rui Santos Nov 24 '15 at 10:30

30

For example, see [this link]() and the snippet below:

```
div#outer {
  height: 120px;
  background-color: red;
}

div#inner {
  width: 50%;
  height: 100%;
  background-color: green;
  margin: 0 auto;
  text-align: center; /* For text alignment to center horizontally. */
  line-height: 120px; /* For text alignment to center vertically. */
}


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

Run code snippet Expand snippet

If you have a lot of children under a parent, so your CSS content must be like this example on fiddle.

The HTML content look likes this:

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

Then see this example on fiddle.

edited Mar 17 '18 at 18:44

community wiki
4 revs, 3 users 63%
Lalit Kumar

27

Centering only horizontally

In my experience, the best way to center a box horizontally is to apply the following properties:

The container:

- should have `text-align: center;`

The content box:

- should have `display: inline-block;`

Demo:

```
  width: 100%;
  height: 120px;
  background: #CCC;
  text-align: center;
}

.centered-content {
  display: inline-block;
  background: #FFF;
  padding: 20px;
  border: 1px solid #000;
}



<div class="container">
  <div class="centered-content">
    Center this!
  </div>
</div>
```

Run code snippet   Expand snippet

See also **this Fiddle**!

Centering both horizontally & vertically

In my experience, the best way to center a box **both** vertically and horizontally is to use an additional container and apply the following properties:

The outer container:

- should have `display: table;`

The inner container:

- should have `display: table-cell;`
- should have `vertical-align: middle;`

The content box:

- should have `display: inline-block;`

Demo:

```
.outer-container {
  display: table;
  width: 100%;
  height: 120px;
  background: #CCC;
}

.inner-container {
  display: table-cell;
  vertical-align: middle;
  text-align: center;
}

.centered-content {
  display: inline-block;
  background: #FFF;
  padding: 20px;
  border: 1px solid #000;
}


<div class="outer-container">
  <div class="inner-container">
    <div class="centered-content">
      Center this!
    </div>
  </div>
</div>
```

Run code snippet   Expand snippet

See also this Fiddle!

26

The easiest way:

```
#outer {
  width: 100%;
  text-align: center;
}
#inner {
  margin: auto;
  width: 200px;
}


<div id="outer">
  <div id="inner">Blabla</div>
</div>
```

Run code snippet   Expand snippet

edited Mar 17 '18 at 18:51

community wiki
4 revs, 3 users 57%
joan16v

1 As your fiddle notes, #inner has to have a width set on it. – Michael Terry Feb 5 '15 at 21:06

`#outer` doesn't need any `width:100%;` as the `<div>` by default always has `width:100%` . and `text-align:center` is also not a necessary at all. – Mobarak Ali May 6 at 22:57

25

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)

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.<br>
          Duis condimentum sem non turpis consectetur blandit.<br>
          Donec dictum risus id orci ornare tempor.<br>
          Proin pharetra augue a lorem elementum molestie.<br>
```

```
</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

24

You can do something like this

```
#container {
   display: table;
   width: <width of your container>;
   height: <height of your container>;
}

#inner {
   width: <width of your center div>;
   display: table-cell;
   margin: 0 auto;
   text-align: center;
   vertical-align: middle;
}
```

edited Oct 16 '13 at 17:08

community wiki
Kenneth Palaganas

24

Here is what you want in the shortest way.

[JSFIDDLE]

```
#outer {
    margin - top: 100 px;
    height: 500 px; /* you can set whatever you want */
    border: 1 px solid# ccc;
}

#inner {
    border: 1 px solid# f00;
    position: relative;
    top: 50 % ;
    transform: translateY(-50 % );
}
```

edited Nov 9 '15 at 3:35

community wiki
2 revs, 2 users 71%
caniz

3 That centers it vertically. – Michael Terry Feb 6 '15 at 0:24

22

Text-align: center

Applying `text-align: center` the inline contents are centered within the line box. However since the inner div has by default `width: 100%` you have to set a specific width or use one of the following:

- `display: block`
- `display: inline`

Show code snippet

Margin: 0 auto

Using `margin: 0 auto` is another option and it is more suitable for older browsers compatibility. It works together with `display: table` .

Show code snippet

Flexbox

`display: flex` behaves like a block element and lays out its content according to the flexbox model. It works with `justify-content: center` .

Please note: Flexbox is compatible with most of the browsers but not all. See here for a complete and up to date list of browsers compatibility.

Show code snippet

Transform

`transform: translate` lets you modify the coordinate space of the CSS visual formatting model. Using it, elements can be translated, rotated, scaled, and skewed. To center horizontally it require `position: absolute` and `left: 50%` .

Show code snippet

`<center>` (Deprecated)

The tag `<center>` is the HTML alternative to `text-align: center` . It works on older browsers and most of the new ones but it is not considered a good practice since this feature is obsolete and has been removed from the Web standards.

Show code snippet

22

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

21

This method also works just fine:

```
div.container {
   display: flex;
   justify-content: center; /* for horizontal alignment */
   align-items: center;     /* for vertical alignment   */
}
```

For the inner `<div>`, the only condition is that its `height` and `width` must not be larger than the ones of its container.

answered Sep 23 '16 at 6:49

community wiki
Billal Begueradj

21

Flex have more than 97% browser support coverage and might be the best way to solve these kind of problems within few lines:

```
#outer {
  display: flex;
```

edited Mar 17 '18 at 18:34

community wiki

2 revs, 2 users 83%
Sandesh Damkondwar

21

Well, I managed to find a solution that maybe will fit all situations, but uses JavaScript:

Here's the structure:

```
<div class="container">
  <div class="content">Your content goes here!</div>
  <div class="content">Your content goes here!</div>
  <div class="content">Your content goes here!</div>
</div>
```

And here's the JavaScript snippet:

```
$(document).ready(function() {
  $('.container .content').each( function() {
    container = $(this).closest('.container');
    content = $(this);

    containerHeight = container.height();
    contentHeight = content.height();

    margin = (containerHeight - contentHeight) / 2;
    content.css('margin-top', margin);
  })
});
```

If you want to use it in a responsive approach, you can add the following:

```
$(window).resize(function() {
  $('.container .content').each( function() {
    container = $(this).closest('.container');
    content = $(this);

    containerHeight = container.height();
```

```
        content.css('margin-top', margin);


      })
    });
```

edited Mar 17 '18 at 18:47

community wiki
2 revs, 2 users 91%
Miguel Leite

20

One option existed that I found:

Everybody says to use:

```
margin: auto 0;
```

But there is another option. Set this property for the parent div. It works perfectly anytime:

```
text-align: center;
```

And see, child go center.

And finally CSS for you:

```
#outer{
     text-align: center;
     display: block; /* Or inline-block - base on your need */
}

#inner
{
     position: relative;
     margin: 0 auto; /* It is good to be */
}
```

edited Mar 17 '18 at 18:38

community wiki

text-align work for text alignment in its container not for its container to its parent. – Lalit Kumar Maurya Dec 4 '13 at 7:32

i test it , i problem with set child to center , must when you have more one child , more times margin:0 auto font answer , but , text-align center , for parent make this child be center , even if they are element and not be text , test and see what happen – Pnsadeghy Dec 4 '13 at 8:35

text-align center text only. You right at this time but when you write a container css which contains a child with different width and color your code does't work. Test it again!!!! – Lalit Kumar Maurya Dec 4 '13 at 9:23

See this example jsfiddle.net/uCdPK/2 and tell me what do you think about it!!!!! – Lalit Kumar Maurya Dec 4 '13 at 10:03

1 2 3 4 next