

# Understanding vertical-align, or "How (Not) To Vertically Center Content"

A FAQ on various IRC channels I help out on is “How do I vertically center my stuff inside this area?” This question is often followed by “I’m using `vertical-align:middle` but it’s not working!”

The problem here is three-fold:

- HTML layout traditionally was not designed to specify vertical behavior. By its very nature, it scales width-wise, and the content flows to an appropriate height based on the available width. Traditionally, horizontal sizing and layout is easy; vertical sizing and layout was derived from that.
- The reason `vertical-align:middle` isn’t doing what is desired want is because the author doesn’t understand what it’s supposed to do, but ...
- ... this is because the CSS specification really screwed this one up (in my opinion)—`vertical-align` is used to specify two completely different behaviors depending on where it is used.

## vertical-align in table cells

When used in table cells, `vertical-align` does what most people expect it to, which is mimic the (old, deprecated) `valign` attribute. In a modern, standards-compliant browser, the following three code snippets do the same thing:

```
<td valign="middle"> <!-- but you shouldn't ever use valign --> </td>
<td style="vertical-align:middle"> ... </td>
<div style="display:table-cell; vertical-align:middle"> ... </div>
```

Shown in your browser, the above (with appropriate wrappers) display as:

<code>&lt;td&gt; using valign="middle"</code>	<code>&lt;td&gt; using valign="bottom"</code>
<code>&lt;td&gt; using vertical-align:middle</code>	<code>&lt;td&gt; using vertical-align:bottom</code>
<code>&lt;div&gt; using display:table-cell; vertical-align:middle</code>	<code>&lt;div&gt; using display:table-cell; vertical-align:bottom</code>

## vertical-align on inline elements

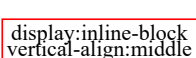
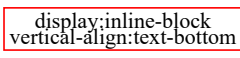
When `vertical-align` is applied to inline elements, however, it’s a whole new ballgame. In this situation, it behaves like the (old, deprecated) `align` attribute did on `<img>` elements. In a modern, standards-compliant browser, the following three code snippets do the same thing:

```
<img align="middle" ...>
<img style="vertical-align:middle" ...>
<span style="display:inline-block; vertical-align:middle"> foo<br>bar </span>
```

In your browser, here’s how the above code renders:

In this paragraph, I have two images— and —as examples.

In this paragraph, I have two images— and —as examples.

In this paragraph, I have a cute little `<span>`  and  as an example.

## vertical-align on other elements

Technically, this CSS attribute doesn’t go on any other kinds of elements. When the novice developer applies `vertical-align` to normal block elements (like a standard `<div>`) most browsers set the value to inherit to all inline children of that element.

## So how *do* I vertically-center something?!

---

If you are reading this page, you're probably not as interested in why what you were doing is wrong. You probably want to know how to do it properly.

### Method 1

---

The following example makes two (non-trivial) assumptions. If you can meet these assumptions, then this method is for you:

- You can put the content that you want to center inside a block and specify a fixed height for that inner content block.
- It's alright to absolutely-position this content. (Usually fine, since the parent element inside which the content is centered can still be in flow.)

If you can accept the above necessities, the solution is:

1. Specify the parent container as `position:relative` or `position:absolute`.
2. Specify a fixed height on the child container.
3. Set `position:absolute` and `top:50%` on the child container to move the top down to the middle of the parent.
4. Set `margin-top:-yy` where `yy` is half the height of the child container to offset the item up.

An example of this in code:

```
<style type="text/css">
    #myoutercontainer { position:relative }
    #myinnercontainer { position:absolute; top:50%; height:10em; margin-top:-5em }
</style>
...
<div id="myoutercontainer">
    <div id="myinnercontainer">
        <p>Hey look! I'm vertically centered!</p>
        <p>How sweet is this?!</p>
    </div>
</div>
```

In your browser, the above example renders as:

Hey look! I'm vertically centered!

How sweet is this?!

### Method 2

---

This method requires that you be able to satisfy the following conditions:

- You have only a single line of text that you want to center.
- You can specify a fixed-height for the parent element.

If you can accept the above necessities, the solution is:

1. Set the `line-height` of the parent element to the fixed height you want.

An example of this in code:

```
<style type="text/css">
    #myoutercontainer2 { line-height:4em }
</style>
...
<p id="myoutercontainer2">
    Hey, this is vertically centered. Yay!
</p>
```

In your browser, the above example renders as:

Hey, this is vertically centered. Yay!