# 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:

1. 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.
2. 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 …
3. … 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:

|  |  |
| --- | --- |
| <td> using valign="middle" | <td> using valign="bottom" |

|  |  |
| --- | --- |
| <td> using vertical-align:middle | <td> using vertical-align:bottom |

<div> using display:table-cell; vertical-align:middle

<div> using display:table-cell; vertical-align:bottom

## 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) alignattribute 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—IMG_256 and IMG_257—as examples.

In this paragraph, I have two images—IMG_258 and IMG_259—as examples.

In this paragraph, I have a cute little <span> display:inline-block  
vertical-align:middle and display:inline-block  
vertical-align:text-bottom 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">

# 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:

1. 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.
2. 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 …
3. … 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:

|  |  |
| --- | --- |
| <td> using valign="middle" | <td> using valign="bottom" |

|  |  |
| --- | --- |
| <td> using vertical-align:middle | <td> using vertical-align:bottom |

<div> using display:table-cell; vertical-align:middle

<div> using display:table-cell; vertical-align:bottom

## 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) alignattribute 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—IMG_256 and IMG_257—as examples.

In this paragraph, I have two images—IMG_258 and IMG_259—as examples.

In this paragraph, I have a cute little <span> display:inline-block  
vertical-align:middle and display:inline-block  
vertical-align:text-bottom 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!

*Copyright ©2004 [Gavin Kistner](mailto:!@phrogz.net), all rights reserved. Comments/suggestions/flames welcome.*

Hey, this is vertically centered. Yay!

</p>

In your browser, the above example renders as:

Hey, this is vertically centered. Yay!

*Copyright ©2004 [Gavin Kistner](mailto:!@phrogz.net), all rights reserved. Comments/suggestions/flames welcome.*

### Method 1

* 假定：
* You can put the content that you want to center inside a block and specify a fixed height for that inner content block.把要居中的内容放在块级元素中，并为该块级元素指定宽度。
* 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.可以为该内容设置绝对定位，因为父元素仍在文档流中。

If you can accept the above necessities, the solution is:如果你可以接受以上两个假设，那么解决办法是：

1. Specify the parent container as position:relative or position:absolute.把父容器设定为绝对定位或相对定位
2. Specify a fixed height on the child container.给子容器设置高度
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.

假设有如下代码结果为（ ）

var emp=new Array(3); emp[1]=1; emp[2]=2; alert(emp[0]);

A 报错

B NaN

C undefined

D 0

哪个不能得到一个学生对象( )

A var student={ }

B var student=new Object( )

C var student=new Student[]

D var student=new Student( )

var  s="abcdefg"   alert (s.substr(1,2)) 结果是？（ ）

1. bc
2. b
3. ab
4. a

[22如果一个字符串str中不含有@，则运行str.indexOf(“@”)后的结果是 (](mailto:22如果一个字符串str中不含有@，则运行str.indexOf(\“@\”)后的结果是() )

A 0

B 1

C -1

D 2