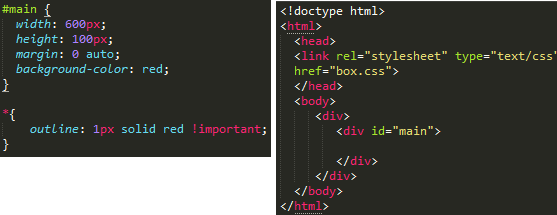
**margin: auto;**

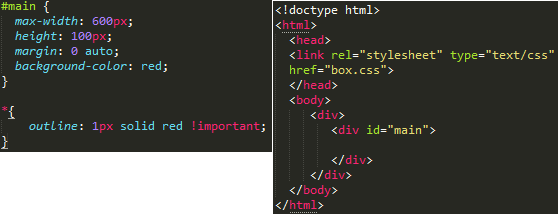
Setting the width of a block-level element will prevent it from stretching out to the edges of its container to the left and right. Then, you can set the left and right margins to auto to horizontally centre that element within its container. The element will take up the width you specify, then the remaining space will be split evenly between the two margins.



**margin**:**0 auto**;

**0** is for top-bottom margin and **auto** for left-right margin. It **means** that left and right **margin** will take **auto margin** according to the width of the element and the width of the container.

The only problem occurs when the browser window is narrower than the width of your element. The browser resolves this by creating a horizontal scrollbar on the page. Let's improve the situation...

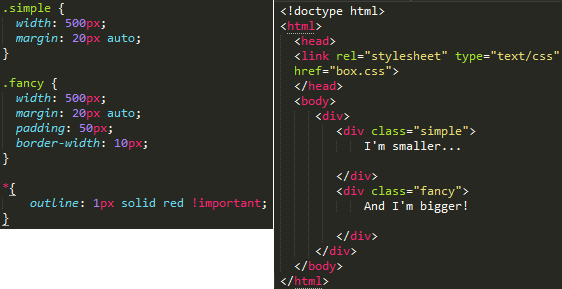


Using max-width instead of width in this situation will improve the browser's handling of small windows. This is important when making a site usable on mobile. Resize this page to check it out!

By the way, max-width is [supported by all major browsers](http://caniuse.com/#search=max-width) including IE7+ so you shouldn't be afraid of using it.

# the box model:

# While we're talking about width, we should talk about width's big caveat: the *box model*. When you set the width of an element, the element can actually appear bigger than what you set: the element's border and padding will stretch out the element beyond the specified width. Look at the following example, where two elements with the same width value end up different sizes in the result.



For generations, the solution to this problem has been math. CSS authors have always just written a smaller width value than what they wanted, subtracting out the padding and border. Thankfully, you don't have to do that anymore...

text-align: center will center the contents of the container, while margin: auto will center the container itself.

You should be careful with margin:auto because it is not supported in IE6. In IE6, you should set text-align:center on the outer container, and reset the CSS for the container itself to text-align:left.

text-align=center used to align the **content** (text for example) to center, however margin: auto is used to align the element itself to center.

no layout:

Having no layout whatsoever is almost ok if all you want is one big column of content. However, if a user makes the browser window really wide, it gets kind of annoying to read: after each line your eyes have a long distance to travel right-to-left to the next line. Try resizing your browser to see what I mean!

Before we fix this problem, let's make sure we're clear on the very important display property.

The visibility property only tells the browser whether to show an element or not. It's either visible (visible - you can see it), or invisible (hidden - you can't see it).

The display property tells the browser **how** to draw and show an element, if at all - whether it should be displayed as an inline element (i.e. it flows with text and other inline elements) or a block-level element (i.e. it has height and width properties that you can set, it's floatable, etc), or an inline-block (i.e. it acts like a block box but is laid inline instead) and some others (list-item, table, table-row, table-cell, flex, etc).

When you set an element to display: block but *also* set visibility: hidden, the browser still treats it as a block element, except you just don't see it. Kind of like how you stack a red box on top of an invisible box: the red box looks like it's floating in mid-air when in reality it's sitting on top of a physical box that you can't see.

In other words, this means elements with display that isn't none will still affect the flow of elements in a page, regardless of whether they are visible or not. Boxes surrounding an element with display: none will behave as if that element was never there (although it remains in the DOM).

# margin: auto;

#main {

width: 600px;

margin: 0 auto;

}

<div id="main"></div>

Setting the width of a block-level element will prevent it from stretching out to the edges of its container to the left and right. Then, you can set the left and right margins to auto to horizontally center that element within its container. The element will take up the width you specify, then the remaining space will be split evenly between the two margins.

The only problem occurs when the browser window is narrower than the width of your element. The browser resolves this by creating a horizontal scrollbar on the page. Let's improve the situation...

# max-width

#main {

max-width: 600px;

margin: 0 auto;

}

<div id="main">

Using max-width instead of width in this situation will improve the browser's handling of small windows. This is important when making a site usable on mobile. Resize this page to check it out!

By the way, max-width is supported by all major browsers including IE7+ so you shouldn't be afraid of using it.