

The CSS Overflow Property

* css-tricks.com/the-css-overflow-property/

Published May 27, 2009 by Chris Coyier

Every single element on a page is a rectangular box. The sizing, positioning, and behavior of these boxes can all be controlled via CSS. By behavior, I mean how the box handles it when the content inside and around it changes. For example, if you don't set the height of a box, the height of that box will grow as large as it needs to be to accommodate the content. But what happens when you do set a specific height or width on a box, and the content inside cannot fit? That is where the CSS overflow property comes in, allowing you to specify how you would like that handled.

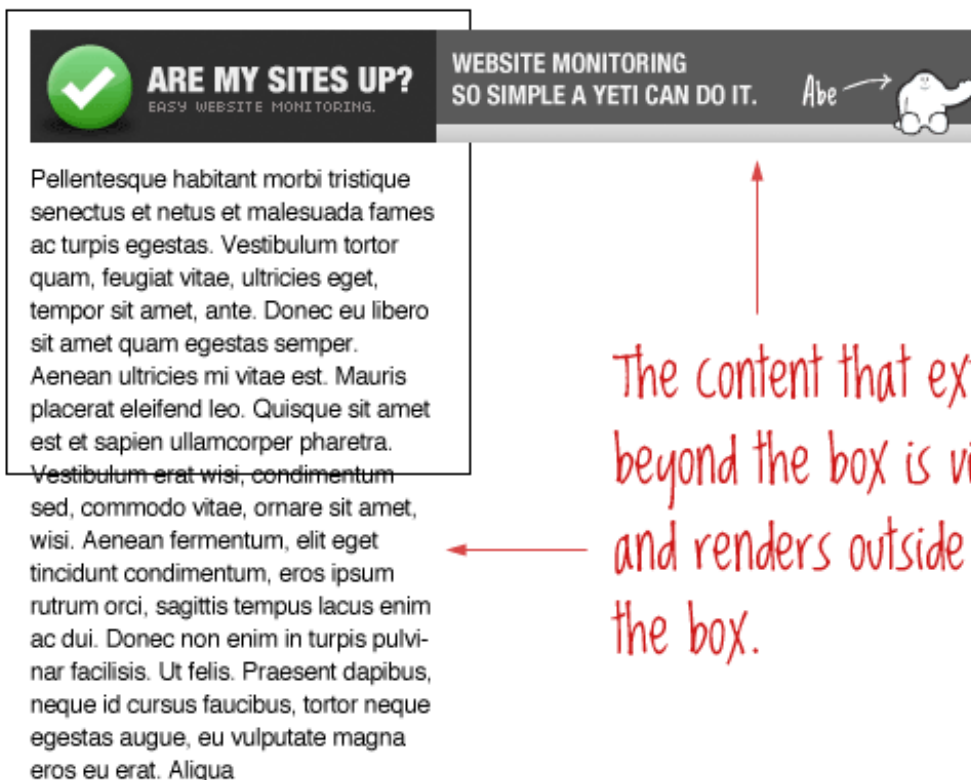
There are four values for the overflow property: **visible** (default), **hidden**, **scroll**, and **auto**. There are also sister properties `overflow-y` and `overflow-x`, which enjoy less widespread adoption.

Let's take a look at each and then discuss some common uses and quirks.

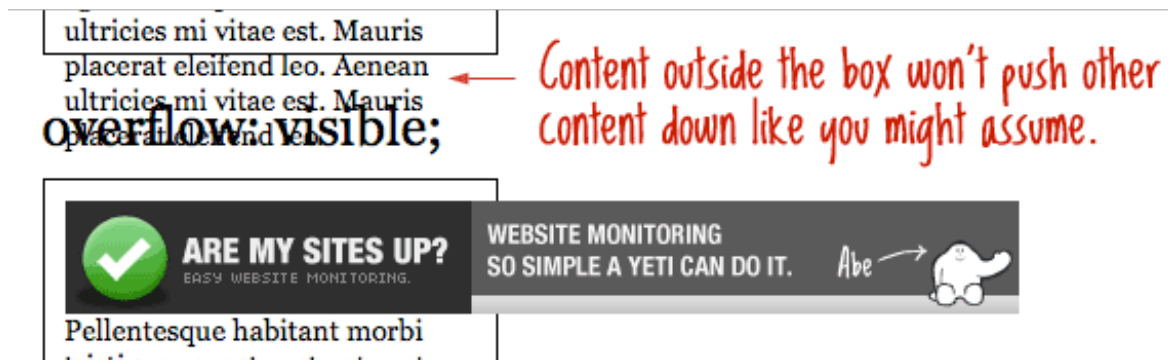
Visible

If you don't set the overflow property at all, the default is visible. So in general, there is no reason to explicitly set this property to visible unless you are over-riding it from being set elsewhere.

```
.box { overflow: visible; }
```



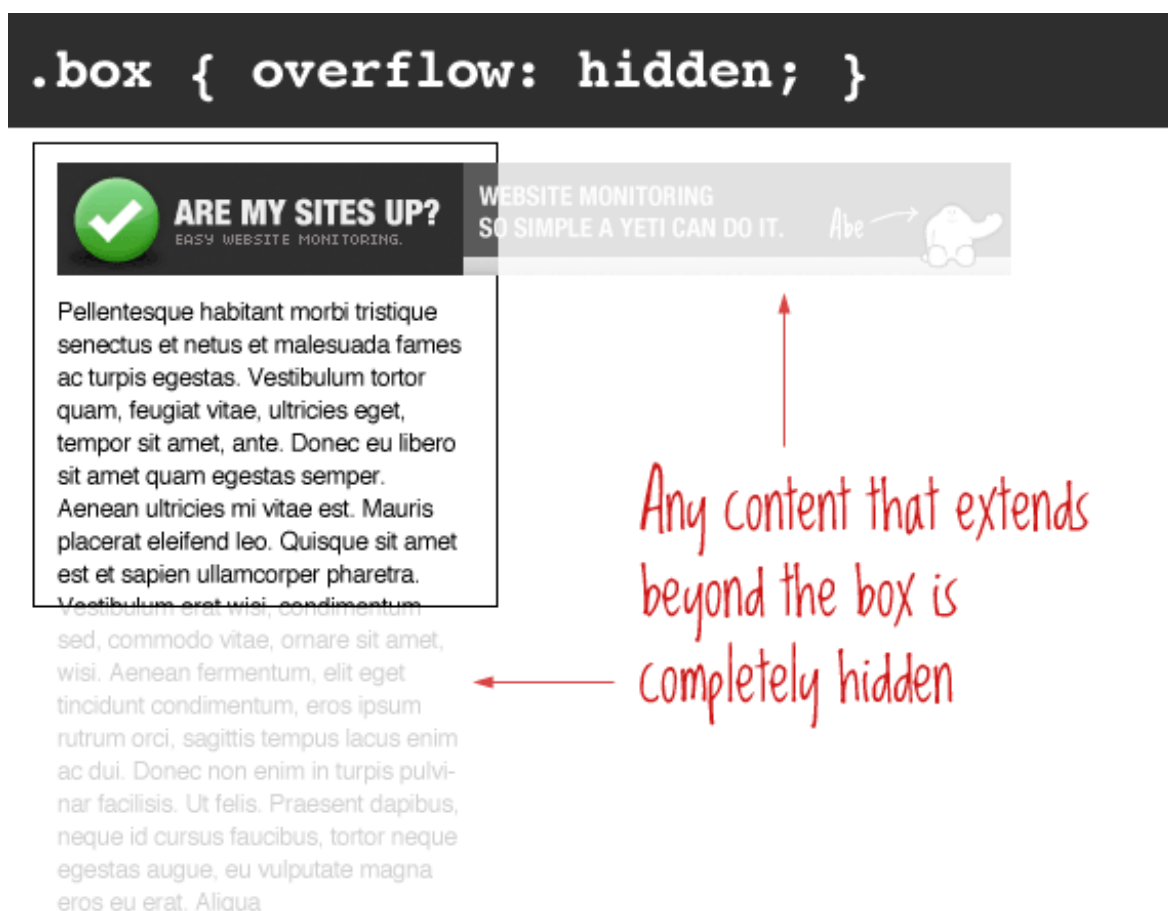
The important thing to remember here is that even though the content is visible outside of the box, that content does not affect the flow of the page. For example:



Generally, you shouldn't be setting static heights on boxes with web text in them anyway, so it shouldn't come up.

Hidden

The opposite of the default visible is **hidden**. This literally hides any content that extends beyond the box.

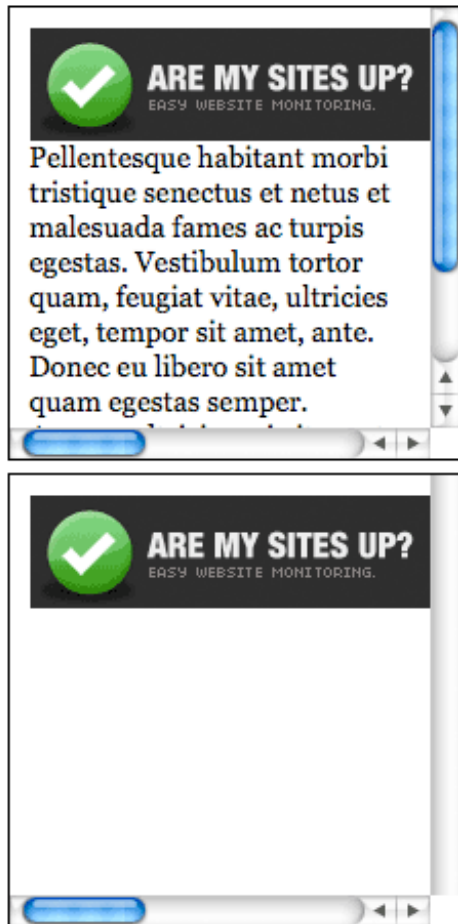


This is particularly useful in use with dynamic content and the possibility of an overflow causing serious layout problems. However, bear in mind that content that is hidden in this way is utterly inaccessible (short of viewing the source). So for example a user has their default font size set larger than you would expect, you may be pushing text outside of a box and hiding it completely from their view.

Scroll

Setting the overflow value of a box to scroll will hide the content from rendering outside the box, but will offer scrollbars to scroll the interior of the box to view the content.

```
.box { overflow: scroll; }
```



Setting overflow to scroll
will give you scrollbars BOTH
horizontally and vertically.
(even if they don't need it)

Of note with this value is that you get BOTH horizontal and vertical scrollbars no matter what, even if the content requires only one or the other.

Auto

Auto overflow is very similar to the scroll value, only it solves the problem of getting scrollbars when you don't need them.

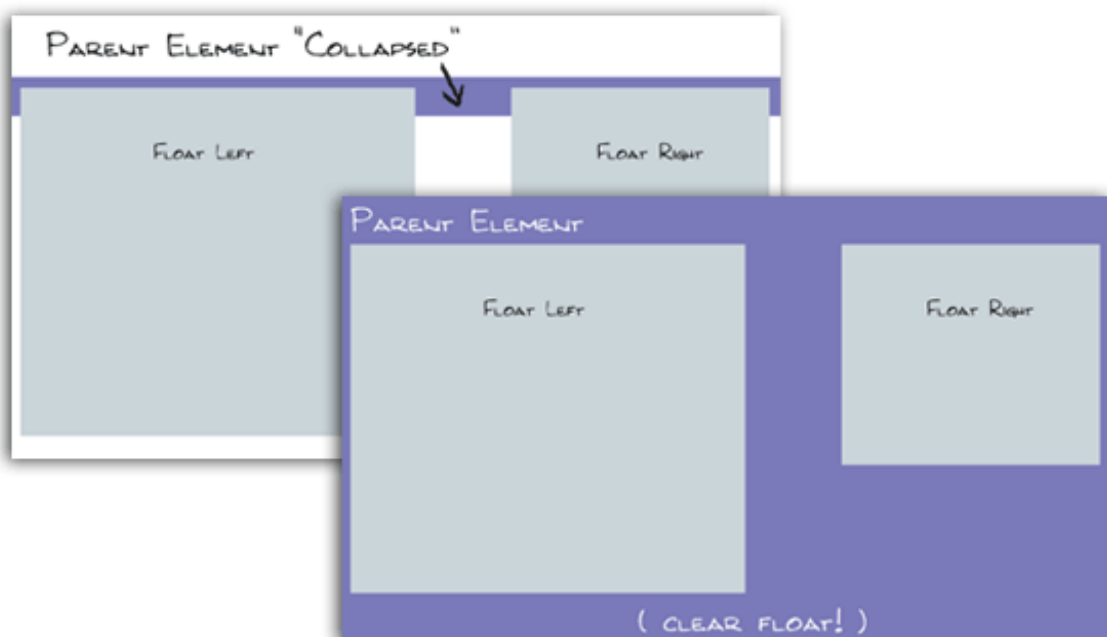
```
.box { overflow: auto; }
```



Auto overflow only gives the box the scroll bars it needs. (even none at all)

Float Clearing

One more popular uses of setting overflow, strangely enough, is float clearing. Setting overflow doesn't clear the float *at* the element, it *self-clears*. This means that the element with overflow applied (**auto** or **hidden**), will extend as large as it needs to encompass child elements inside that are floated (instead of collapsing), assuming that the height isn't declared. Like this:



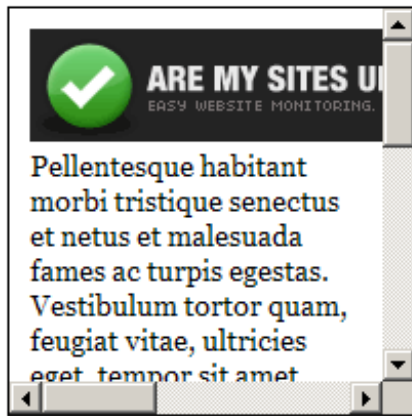
There is more detail about this and floats in general in the article [All About Floats](#).

Cross Browser Concerns

Like most things in CSS, there are plenty of cross-browser quirks with overflow. Here they are:

Scroll bars inside or outside of box?

Firefox puts them outside, IE puts them inside. I believe only IE actually gets this correct (it should be inside).



IE 7



Firefox 3.0.10

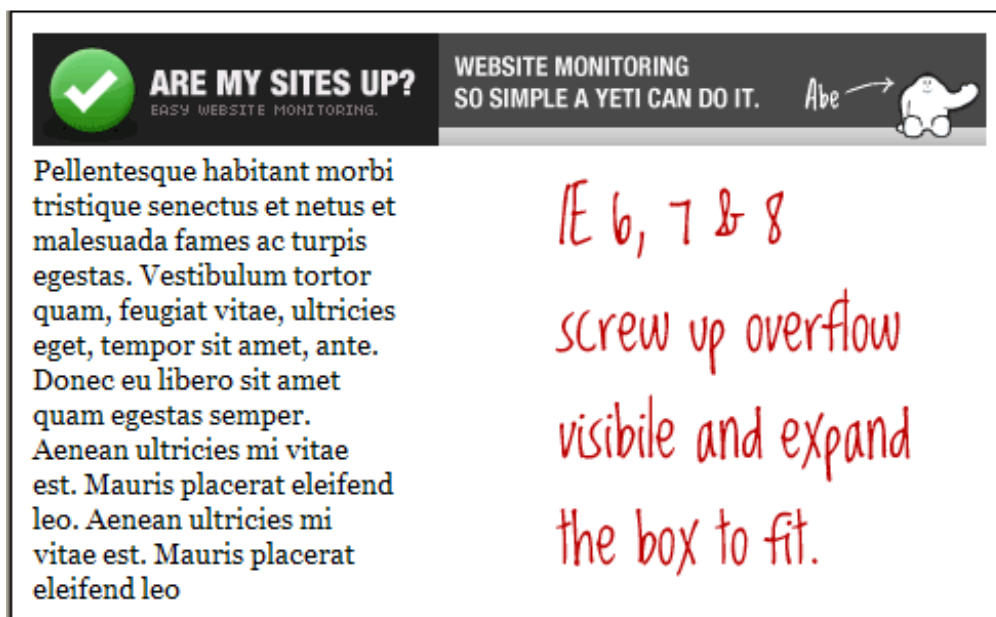
Make sure to account for this substantial difference.

IE 8 expanding box bug

There are lots of fun new bugs, including some very serious ones that hide entire web pages, with IE. [More details here.](#)

Breaking floated layouts

IE 6, 7 and 8 all screw up the default overflow visible and will expand a box horizontally to fit content (likely an image). This is especially painful for layouts built on floated columns, and a single expanded column can push other columns down and cause some pretty seriously borked layouts.



IE 6, 7 & 8

screw up overflow
visible and expand
the box to fit.

Can scroll bars be styled with CSS?

IE [used to allow this](#) in older versions of IE but it has since been discontinued. WebKit/Blink browsers allow it and [here's some information about that](#). It's non-standard though and it's unclear if the standards bodies are interested in standardizing it. If you absolutely need customized cross-browser scrollbars, you'll need to handle it custom with JavaScript. Here's [a simple start](#) and here's [a StackOverflow thread](#) with recommendations.

IE Trick

IE displays a vertical scrollbar all the time whether it needs it or not. This can be nice for [preventing horizontal jumps](#), but isn't always desirable. To remove this in IE, you can set overflow to auto on the body element.

Demo

Demos for this article taken from [this sample page](#).