* The html and body elements are distinct block-level entities, in a parent/child relationship. The html element's height and width are controlled by the browser window. It is the html element which has (by default) overflow:auto, causing scrollbars to appear when needed.
* The body element is (by default) position:static, which means that positioned children of it are positioned relative to the html element's coordinate system.

html是body的父级，html的宽高由浏览器窗口决定（确切地说，应该是由viewport决定，即不包括滚动条、工具栏），它默认有overflow:auto属性。body有默认8px的margin，html什么都没有。

* In almost all modern browsers, the built-in offset from the edge of the page is applied through a margin on the body element, **not** padding on the html element.

几乎所有的浏览器，默认的偏移margin是在body元素上，而不是html的padding。所有重置页边距用body {margin:0;padding:0} 缺一不可。

## **Introduction**

Many web developers do not understand the difference between applying style to the body element versus the html element. Most of the time these authors will apply style only to the body element; when that's not sufficient, they'll spam all sorts of styles on both html and body until the page happens to look correct.

The confusion is understandable. In the beginning, both were treated similarly, with (now-deprecated) attributes like bgcolor being applied to the body tag, affecting the whole page.

起初html和body的某些属性是相似的（现在弃用了），比如给body加背景颜色的话会应用到整个页面上。

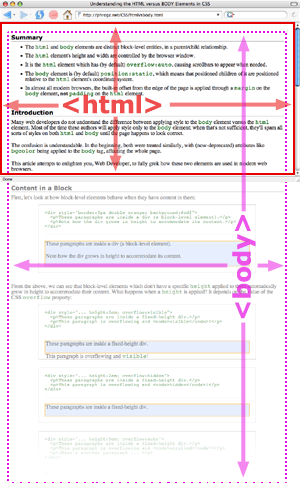
This article attempts to enlighten you, Web Developer, to fully grok how these two elements are used in modern web browsers.

## **Mommy, Where Do Scrollbars Come From?**

On any non-trivial HTML page, there's enough content that the page gets a scrollbar on the side. Where does this scrollbar come from? Magic? An unholy tryst between UI widgets? Maybe. Or maybe it's the simple result of an implicit CSS rule like:

html { overflow:auto }

"But the html element doesn't have a height!" you cry, confused. "How can it generate scrollbars, if it's a block-level element without a specific height?"

Thehtml**element is a little bit special. Its height and width are governed by the window/frame it's in.** When you increase your window width, the fixed width of the html element is increased; when you make the window taller, so increases the height of the html element.

Because pictures often help, let's see what it would look like if the html element could have overflow:visible:

Hopefully this picture helps, rather than confuses. See how the html element (in red) goes to the edges of the window, but the body overflows that container? The html element is responsible for showing the scrollbar for the 'page' when the content gets too tall.

So What?

What does this mean for you? A couple things:

If you're one of those designers who likes fixed-width, centered pages, you can do this by setting a width on the body tag and centering it. ([Example](http://phrogz.net/css/fixedwidthbody.html))

如果你喜欢页面宽度固定并居中，就给body设置宽度并居中。

Because body isn't positioned (or more accurately, it's position:static by default), setting a child element to height:100% causes it to be 100% of the height of the html element, not the body element. Thus, if you want something to be as tall as the body is (going down past the bottom of the page) use body { position:relative }

body是静态定位的，它的后代如果有定位，参照物就是html，所以如果你想让某个元素和页面一样高，就要给body添加 {position:relative}。

(However, see also [Tall Nav](http://phrogz.net/css/tallnav.html) and [Tall Content](http://phrogz.net/css/valign_in_body/block.html).)

Hopefully it helps you understand exactly where margin and padding and border will be appear when applied to the html and body elements.

《css权威指南指出》height / width百分数是相对于包含块的。如果没有显式声明包含块的height，百分数高度会重置为auto（由内容撑开）。