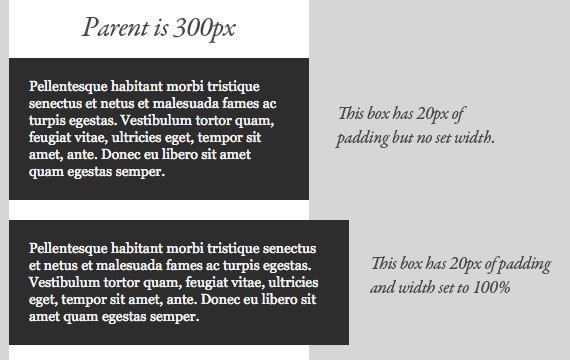
**The Default Width of Block Level Boxes**

If you don’t declare a width, and the box has static or relative positioning, the width will remain 100% in width and the padding and border will push inwards instead of outward. But if you *explicitly set* the width of the box to be 100%, the padding will push the box outward as normal.



The lesson here being that the default width of a box isn’t really 100% but a less tangible “whatever is left”. This is particularly valuable to know, since there are lots of circumstances where it is immensely useful to either **set** or **not set** a width.

The biggest ass-biter I always find with this is textarea elements, which very much need their widths set to fight the required “cols” attribute, and are unable to have children elements. So you often need the textarea to be explicitly set to 100%, yet have padding as well, pushing it too large to fit. In a static width environment, we often resort to pixel widths that fit, but no such luck in fluid width environments.

### Absolute Boxes with No Width

Absolutely positioned boxes that have no width set on them behave a bit strangely. Their width is only as wide as it needs to be to hold the content. So if the box contains a single word, the box is only as wide as that word renders. If it grows to two words, it’ll grow that wide.



This should continue until the box is 100% of the parent’s width (the nearest parent with relative positioning, or browser window) and then begin to wrap. It feels natural and normal for boxes to expand vertically to accommodate content, but it just feels strange when it happens horizontally. That strange feeling is warranted, as there are plenty of quirks in how different browsers handle this, not to mention just the fact that text renders differently across platforms.

