

How CSS works: an explanation for web designers

CSS is not Photoshop. Period. CSS is a style sheet language designed to allow you to build complete page layouts starting from an HTML underlying structure. With CSS, you should rethink your approach to visual design. First, with CSS your design is not a full-box where the end user has no access but rather an interactive field where user's and designer's needs meet each other.

A user can write his/her own CSS to radically change your design. This is called user style sheet and it's a feature supported by all major browsers. This implies a simple fact: when I said that CSS is not Photoshop I meant what I said, in the sense that yes, CSS can be used for visual design, but CSS requires that you should first forget the way by which you usually create a template and then start thinking in a more dynamic way.

Photoshop is static: once a PSD file is turned into an image there's no turning back. It's done. Instead, a CSS layout is always dynamic in the sense that it changes following certain user actions. For example, did you notice what happens to a pixel-perfect layout when you resize the font size? Kaboom! Everything falls apart, especially when the designer didn't take into account this case.

That's why CSS gurus stick to fluid and elastic layouts: they allow you to scale along with certain user interactions, such as a window resize or a font resize. The special length unit called em doesn't exist in Photoshop. In fact, you have pixels, points and so on but no ems. This unit depends on the font family in use. More specifically, an em is the height of the lowercase 'm' letter of the font in use, so 1em in Arial is different from 1em in Georgia.

Don't worry: some astute CSS developers have found a special constant to set 1em equal to 10 pixels:

```
body { font-size: 62.5%;}
```

So if you have a box that must be 120 pixels wide, with this constant you can use 12em. Instead, percentages require a further step away from Photoshop. In Photoshop you have all kinds of lengths, except ems (as we saw earlier) and percentages. Now follow me: percentages refer to the width of the parent element. For example:

```
#parent {  
    width: 300px;  
}  
  
#child {  
    width: 20%;  
}
```

If my calculator is right, the 20% of 300 pixels is 60 pixels, so our child element will be 60 pixels wide. Things tend to get more complicated when every element on the page has its dimensions expressed in percentages. In this case, browsers will calculate the values taking the viewport width as reference. So if your viewport is 1600 pixels wide, you get a different result for 20%. Again, with another screen resolution you'll get another result in pixels.

Finally, CSS is an interpreted language. A CSS layout is first interpreted and then displayed. The problem is that not all browsers interpret CSS in the same way. On the contrary, a Flash movie will be always interpreted in the same way because there's only a single interpreter. Firefox, Safari, Chrome, Opera and Internet Explorer are all CSS interpreters that try to render CSS

according to the specifications. Sometimes they get it right, other times it's all wrong. But that's the point: with CSS you have the benefit of surprise.