

Common CSS Properties

Intro

As mentioned before there are a lot of CSS properties. Some vary from browser to browser. Some are only in the spec but are not widely supported. As a web developer you will need to get comfortable looking through lists of properties looking for one that might do what you want. The other thing you will want to do is when you see an interesting styling on a website, inspect it and see what properties they used to do it.

All of that said there are a few properties you are going to be using all the time and some of them are not entirely intuitive and those properties are what we will be discussing in this section.

Colors

Colors in CSS are represented in a couple ways. One is hex values in the form of `#RRGGBB`. Larger R values give more red, G green and B blue. So `#FF0000` would be pure red. `#FFFFFF` would be white. `#550055` would be a darkish purple. In addition to using hex there are several named colors. A list of the basic named colors and their corresponding hex values can be found here (<http://www.w3.org/TR/css3-color/#html4>). Personally when I am just using colors to help separate things for layout I use named colors. When actually designing a sites color scheme I use hex colors for more control.

Some Example Color Properties

color

The color of the font in an element.

border-color

The color of the border of an element

background-color

The background color of an element

Positioning

CSS is often used to position elements. When we talk about the position of an element we need to say what it is in relation to. In the real world we might say that a treasure is buried at 56 degrees N and 12 W. Or we might say it is 300 miles West of Glasgow. In the same way in CSS we can talk about where an item is relative to the screen or relative to its ancestor element.

Static

Static positioning is the default positioning. The browser will figure out where something ought to be using the default algorithm. As a designer you don't have a lot of control here.

Relative

Objects with relative positioning are placed where they normally would be then the change in their location is measured from that spot.

Absolute

This sets the position based on the parent element. If an elements absolute position is set to 0 by default the top left corner of the element be in the exact same position as the top left corner of its parent element.

Fixed

This fixes the position of the element with respect to the document window. If you scroll the page the fixed element will scroll with the page. So you can make a nav bar always be at the top of the page. I would suggest not using fixed placement of elements as it can make for a difficult user experience on smaller screens.

Measurements

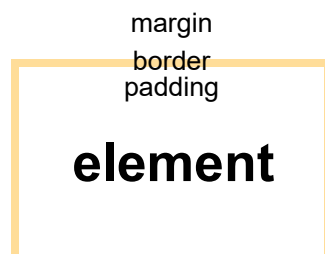
We know how to position things but what about actually measuring distances? In CSS there are a lot of options for measurement. We are only going to look at two `px` and `em`.

`px` is an absolute measurement in pixels. We are going to assume a pixel is equal to one pixel on a display. Sometimes this isn't true. We are going to pretend it is and move on.

`em` is a relative measure based on the font size of a page. This is a nice tool to use for layouts because if a user increases their font size to make a site easier to read the size of elements will expand as well. If you were to use a fixed measurement and a user increased the font size text might spill outside of the elements which are supposed to contain it. Finally `%` can be used which is a measurement of the parent. So setting an element to `width:50%;` will make it half as wide as its parent element.

Spacing

Finally we are going to talk about spacing. When laying out documents essentially everything is a box. Inside every box is the content of the element. When laying out these boxes we can specify two things *how far one box is from another* and *how much space there is between the edge of a box and its content*. We refer to these as margin and padding respectively.



From this position the sides `left` `right` `top` `bottom` are used to move the element. `left: 5px;` means that the left edge of the box will be offset away from its normal position moving the element 5px right. `top: 10px;` means the top of the element will be 10 pixels from where it normally is shifting the element down 5 pixels.



Activity

Go to this fiddle (<http://jsfiddle.net/wolfordj/1nycqj7/5/>). In the upper right box you can modify the CSS. Play around by setting different `position` properties and setting different `top` or `left` values to move the divs around.

Review

This sub-module is designed to help you learn the basics of positioning elements. There is a lot more to learn beyond what was introduced here but this should let you achieve almost any layout even if there are cleaner ways to do it with more advanced features.