

Shorthand CSS

Have you ever seen a CSS stylesheet with multiple attributes applied to the one property and wondered how it's done? Let's go through it and make your stylesheets that much better. By looking through this article you will not only save time styling your website, you will also make sure the page is a lot faster loading – your website is judged within the first five seconds of someone visiting it.

Most of what will be explained within this article is useful for not only the current version CSS, but the new version, CSS3, so this is useful no matter which version you're currently using in your website development.

Contents

1. [Shorthand properties introduction](#)

You will learn what the term shorthand means and how to use it within your next CSS design. This is very useful to learn the lingo of a CSS designer.

2. [Attributes with shorthand properties](#)

By looking through this section you will take note of the most common attributes that have shorthand properties for adjusting your page layout technique.



Shorthand properties introduction

Shorthand: the art of using a shortcut to apply CSS attributes within the one property.

Picture this. You're there coding a stylesheet and wish to develop a border for an element included in your main document. There are several ways of doing this from making sure you use the separate border tags - `border-bottom`, `border-top` - or you could do this by applying all of the attributes to the one property.

Typically, all shorthand properties are very similar. Many of the commonly used properties will include shorthand alternatives to make sure they can still be used without having to write numerous lines of code to end up with the same result.

Examples of these properties can be seen within the next chapter. Scroll to the next page for that.

The best way to develop your shorthand guide is to actually put your fingers to action in a coding application - code view only please! As there is no official version to go by, make this your alternative guide until you have memorised how to do it offhand.

For further reading, check out CSS cheat sheets. There are numerous versions of these documents, as, again, there is no official version. W3C set the guidelines, take what you will from them.

Attributes with shorthand properties

Now you know what shorthand code is, let's go ahead and play with a few examples. Obviously there are more than what will be covered in this document, but these are the essentials - remember these and you have basically mastered the art of using shorthand coding.

So, for the first ever shorthand version of coding we will be using the 'background' property. Let's go ahead and take the code before we learnt how to use the shorthand coding.

```
body {  
    background-color: #fff;  
    background-image: url('images/background.jpg');  
    background-repeat: no-repeat ; /* repeat-x, repeat-y, no-repeat*/  
    background-attachment: fixed; /* scroll fixed */  
    background-position: center; /* top, bottom, center, left, right,  
center */  
}
```

It looks a lot for just styling the background, doesn't it? Let's change it into one property. We will be using 'background' to apply all of these properties in just the one.

```
body {  
    background: #fff url('images/background.jpg') no-repeat fixed  
center;  
}
```

Done. So how was it done? When you use the 'background' property, what you're actually using is all of the other background properties in the one. This tells the web browser to read the property for the information in this order:

```
background-color | background-image | background-repeat |  
background-attachment | background-position
```

It's easy when you know how, right? Let's move on to margins and padding. These elements are very similar, so once the margin has been described, the same can



be applied to the padding property.

For an example, let's say we have a 10px, 20px, 30px and 40px margin applied at the top, bottom, right and left of the element respectively. Your code would look very similar to shown below.

```
body {  
    margin-top: 10px;  
    margin-right: 20px;  
    margin-bottom: 30px;  
    margin-left: 40px;  
}
```

That's good, if you want to have to type out margin four times Let's change it.

```
body {  
    margin: 10px 20px 30px 40px;  
}
```

That's even better. We now have margins of 10, 20, 30 and 40 pixels applied to the top, right, bottom and left respectively. This done due to the shorthand nature of both the 'margin' and 'padding' property. How does it work? Well, the web browsers look for the attributes in this order.

```
{ top right bottom left } | { top right/left bottom } |  
{ top/bottom right/left } |  
length | percentage
```

As you can see, this is read by using the top, right, bottom, left values. However, if you wish to apply the same margin or padding to the right and left, you can do this by using '10px 20px 30px'. By doing this, it will apply a 10px margin to the top, 20px to the right and left and 30px to the bottom. We can also do something very similar for a same margin or padding to the top and bottom and right and left. Let's have a look. What will '10px 20px' do? It will apply 10px margin or padding - depending on which you used - to the top and bottom as well as 20px to the right and left. The length and percentage attributes count for whether you wish to use length in pixels, ems or similar measurements, or a percentage.

Hopefully now you have more of an insight into shorthand properties. Research more the different properties of CSS which allow for shorthand. It's more much beneficial to you, as a coder, and a web browser as it reduces loading time doing this for all of the properties allowing use of shorthand.