

The Background Shorthand Property

Shorthands exist to make declarations faster and easier. The background shorthand isn't left out too. In this lesson, we will take a look at how the CSS background shorthand really works

The background shorthand property like other shorthands, can be a life saver. When you have a lot of background declarations, it easier to have them declared in one statement.

For example, consider the code below:

```
.bg {  
  background-image: url('path/to/image.png');  
  background-position: 50% 50%;  
  background-size: cover;  
  background-repeat: no-repeat;  
}
```

Using the `background` shorthand, it becomes this:

```
.bg {  
  background: url('path/to/image.png') 50% 50%/cover no-repeat;  
}
```

Isn't that awesome?

How it Works

As seen in the example above, the `background` property takes in background properties and sets them in one declaration.

The background properties include, `url` `background-position` `background-size` `background-repeat` `background-origin` `background-clip` and `background-`

Oh my god, what a long list!

Must I Follow an exact sequence of properties?

The values of the individual background properties in the background shorthand can be shuffled as you deem fit.

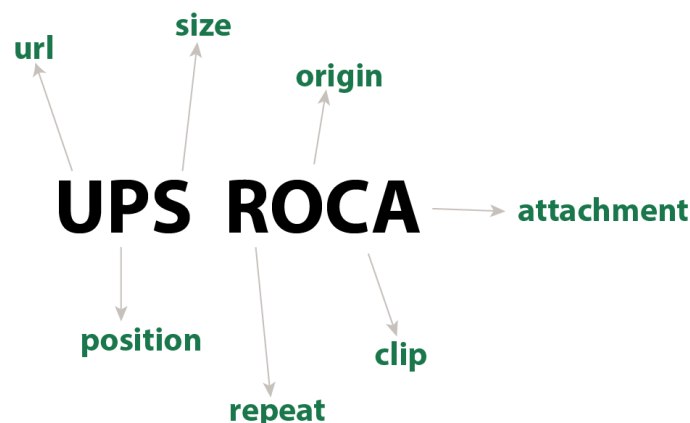
The one rule to always remember is that the `background-size` property must be specified after the `background-position` property. They must also be separated with the `/` character.

You cannot specify the background size in the shorthand property unless you also specify the background position, otherwise the declaration is invalid.

My Personal Recommendation

I do have a sequence I follow - just to keep me sane. I have an acronym that has helped me too. Here it is: **UPS ROCA**

What?



UPS is a global package delivery company. It is easier for me to remember

that. A **ROCKER** (my spelling, **ROCA**) is a person who performs, dances to, or enjoys rock music.

Bring these two together and I imagine a **ROCKER** dancing on a UPS delivery van! I can never forget that. The acronym helps me remember my flavored sequence for the background properties.

But what is **background-origin** **background-clip** and **background-attachment**?

We haven't taken a look at these properties. You need not concern yourself with these for now. Hopefully, I will touch on them in a later practical section.

Remember

1. If you have to specify **background-position** and **background-size** properties, be sure to separate them with a forward slash (/) like this:
background-position / background-size e.g **50% 50% / cover**
2. Any background property NOT set in the shorthand will be replaced with its default value. e.g if you skip **background-repeat**, it will be implicitly replaced with the value **repeat**. Don't want it to repeat, then set the value explicitly.

Exercise

This exercise should help you practice the use of the shorthand **background** property.

The following is the style applied on an element.

```
element {
```

```
.element {  
  background-color: #eee;  
  background-position: top 50%;  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: contain;  
  background-origin: padding-box;  
  background-image: url(path/to/image.jpg);  
  background-clip: border-box;  
}
```

Re-write this declaration to use the **background** shorthand property. If you may, use the acronym UPS ROCA.

CSS (SCSS)

```
.element {  
  background-color: #eee;  
  background-position: top 50%;  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: contain;  
  background-origin: padding-box;  
  background-image: url(path/to/image.jpg);  
  background-clip: border-box;  
}
```



 Show Hint

Now let's finish off the movie project.