While declaring the a solid color uses background-color property in CSS, gradients use background-image. This comes in useful in a few ways which we'll get into later. The shorthand background property will know what you mean if you declare one or the other.

gradient {

/\* can be treated like a fallback \*/

background-color: red;

/\* will be "on top", if browser supports it \*/

background-image: linear-gradient(red, orange);

/\* these will reset other properties, like background-position, but it does know what you mean \*/

background: red;

background: linear-gradient(red, orange);

}

[**#**](https://css-tricks.com/css3-gradients/#article-header-id-1)**Linear Gradient**

Perhaps the most common and useful type of gradient is the linear-gradient(). The gradients "axis" can go from left-to-right, top-to-bottom, or at any angle you chose.

Not declaring an angle will assume top-to-bottom:

.gradient {

background-image:

linear-gradient(

red, #f06d06

);

}

Those comma-separated colors can type of color you normally use: Hex, [named colors](https://css-tricks.com/snippets/css/named-colors-and-hex-equivalents/), [rgba](https://css-tricks.com/rgba-browser-support/), [hsla](https://css-tricks.com/yay-for-hsla/), etc.

To make it left-to-right, you pass an additional parameter at the beginning of the linear-gradient() function starting with the word "to", indicating the direction, like "to right":

.gradient {

background-image:

linear-gradient(

to right,

red, #f06d06

);

}

This "to" syntax works for corners as well. For instance if you wanted the axis of the gradient to start at the bottom left corner and go to the top right corner, you could say "to top right":

.gradient {

background-image:

linear-gradient(

to top right,

red, #f06d06

);

}

If that box was square, the angle of that gradient would have been 45°, but since it's not, it isn't. If you wanted to make sure it was 45°, you could declare that:

.gradient {

background-image:

linear-gradient(

45deg,

red, #f06d06

);

}

You aren't limited to just two colors either. In fact you can have as many comma-separated colors as you want. Here's four:

.gradient {

background-image:

linear-gradient(

to right,

red,

#f06d06,

rgb(255, 255, 0),

green

);

}

You can also declare where you want any particular color to "start". Those are called "color-stops". Say you wanted yellow to take up the majority of the space, but red only a little bit in the beginning, you could make the yellow color-stop pretty early:

.gradient {

height: 100px;

background-color: red;

background-image:

linear-gradient(

to right,

red,

yellow 10%

);

}

We tend to think of gradients as fading colors, but if you have two color stops that are the same, you can make a solid color instantly change to another solid color. This can be useful for declaring a full-height background that simulates columns.

.columns-bg {

background-image:

linear-gradient(

to right,

#fffdc2,

#fffdc2 15%,

#d7f0a2 15%,

#d7f0a2 85%,

#fffdc2 85%

);

}

#### Browser Support / Prefixes

**So far we've only looked at the new syntax,** but CSS gradients have been around for quite a while. Browser support is good. Where it gets tricky is syntax and prefixing. There are **three** different syntaxes that browsers have supported. This isn't what they are officially called, but you can think of it like:

1. **Old:** original WebKit-only way, with stuff like from() and color-stop()
2. **Tweener:** old angle system, e.g. "left"
3. **New:** new angle system, e.g. "to right"

And then prefixing as well.