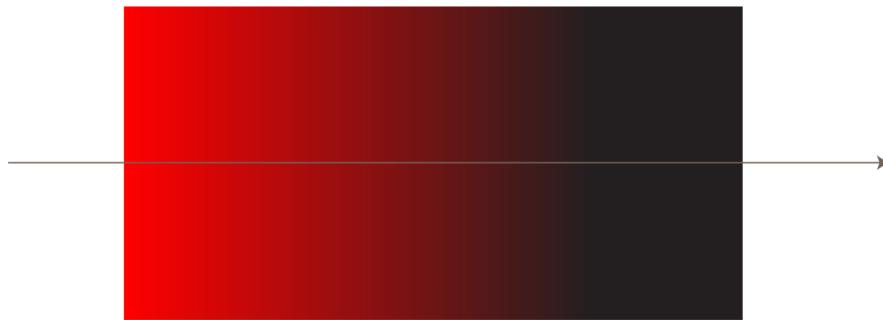# How Linear Gradients really Work

In the previous lessons, you had a feel for how linear gradients work. In this lesson, we will go even deeper

At the start of this section, we defined a linear gradient as a gradient that fades from one color to another **over a line.**

Actually, **a gradient line exists on every linear gradient.** This gradient line serves one purpose, to determine the direction in which the gradient will be painted inside an element.

What's the default direction?

The default starting position of the gradient line is from top to bottom as seen below:

Angle
Measured from the clockwise direction
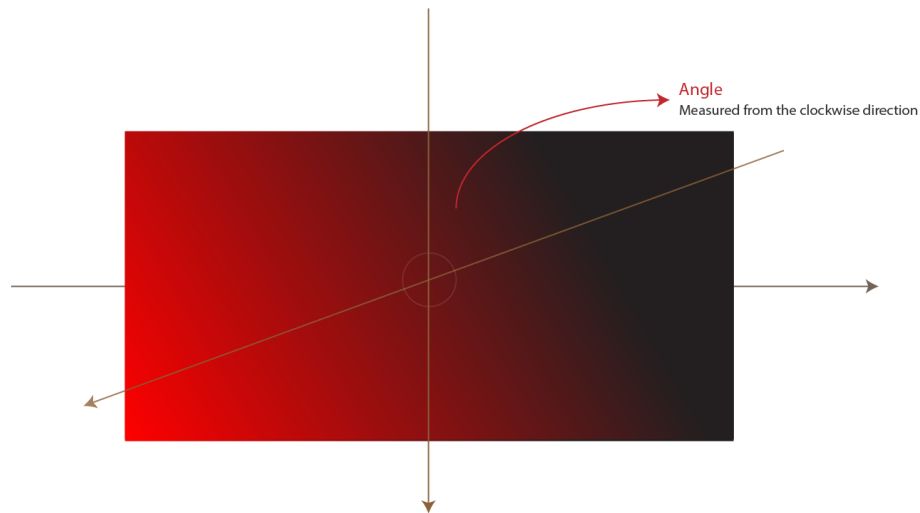
While `top to bottom` is the default, it is also possible to have an angled gradient line. What I mean by that is a gradient line that isn't vertically inclined but tilted at an angle.

Please refer to the graphic above, where I have drawn a gradient line with the angle measured from the clockwise direction.

Let's see an example.

Consider a `div` with a linear-gradient applied to it like so: `background-image: linear-gradient(red, black)`

As you may expect, the result of this is a gradient that blends from top to bottom (the default behaviour)

Check the result below:

HTML   CSS   Output

```css
1
2
3
4
5   .grad {
6       background-image: linear-gradient
7       width: 100vw;
8       height: 100vh;
9   }
10
11
12
13
```

```
14
15
16
17
```

Gradient Directions

The `direction` of a gradient line isn't set in stone, and may be changed. We will take a look at how this works later in this lesson.

For now, here's what specifying a gradient direction looks like:

```
#element {
    background-image: linear-gradient(angle/direction, yellow, red)
}
```

Do not let the `angle/direction` bit get you confused. It's easier than you think and I take the time to explain below.

Angled Gradient Lines
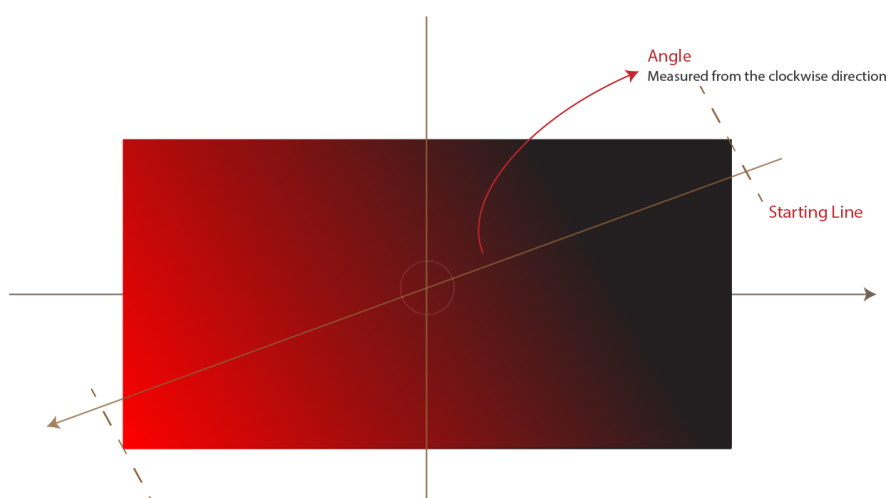
The image above shows the gradient line at an angled position.

The gradient image is painted with lines **perpendicular to the gradient line.** To re-iterate, the gradient line is the line along which the gradient will be painted.

Look at the **starting** and **ending** line in the illustration above. They represent the direction of the perpendicular lines for which the gradient image is painted.

> **The most visual way to process this information is to assume there are *perpendicular* lines running through the gradient line. These lines hold the needed color accross the gradient.**

Note that, every point on the starting line will always be the gradient pure starting color. Every point on the ending line will always be the gradient pure ending color too. The intermediate colors between the first and last are then painted.

For more control, the first step to creating a linear gradient is to specify the gradient line.

Specifying the Gradient Line

The gradient line can be specified in 2 ways. Whichever way you chose, the general syntax looks like this:

```
#element {
    background-image: linear-gradient(angle/direction, yellow, red)
}
```

Let's discuss these two ways.

1. Using Directions

You can specify a gradient line using keywords like:

```
#element {
    background-image: linear-gradient(to top, yellow, red)
}
```

The available keywords include: `to top`, `to bottom`, `to right` and `to left`.

Exercise

Here's a playground for you to try your hands at. Make sure to use the 4 keyword combination above. Also note how they affect the gradient.

HTML   CSS   Output

```
1
2
3    .grad {
4        background-image: linear-gradient(
5        /*edit the line above to specify the
6        width: 100vw;
7        height: 100vh;
8    }
9
10
11
12
13
14
15
16
17
```

One more thing...

There are a few more keywords. I know I should have said that earlier :(

`to top` , `to bottom` , `to right` and `to left` specify directions to a certain side of the element. **To specify corners** `to top left` , `to top right` , `to bottom right` , and `to bottom left` are also possible values.

Be sure to try out these values in the playground above too.

Adding Gradient Color Stops

A gradient with two colors, start and end. Is that all we can do with CSS? Certainly Not.

With color stops, we can add an infinite number of colors within the gradient.

The general syntax looks something like this:

```
linear-gradient(angle/direction, color stop, color stop, ...);
```

In the syntax above, `color` refers to any color value, and `stop` refers to the position of the color i.e where the color begins, measured from the start of the gradient.
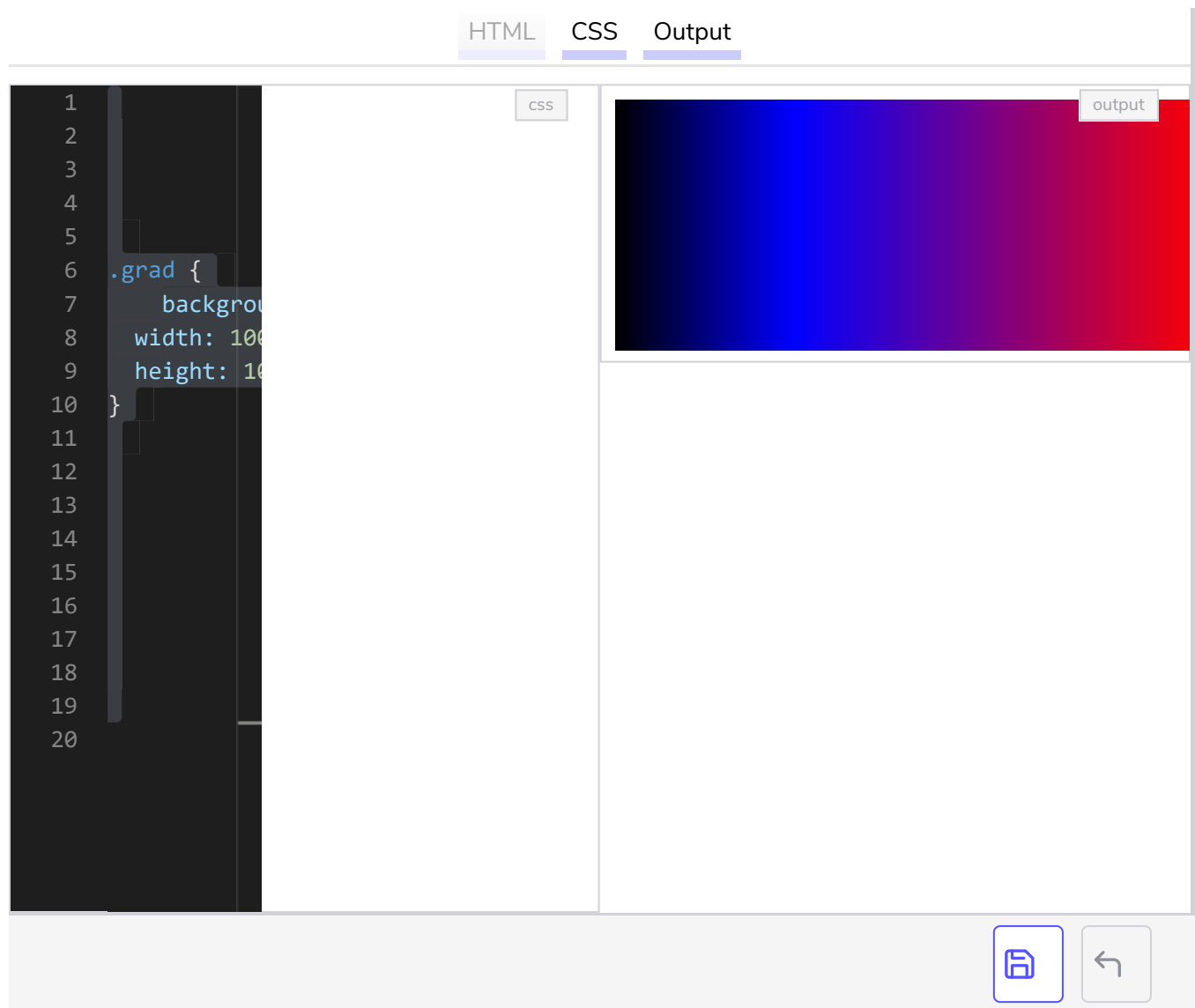
The color stop position can be set in percentage values or fixed length values such as px or em, etc.

For example,

```
#el {
    background-image: linear-gradient(to right, black, blue 30%, red)
```

```
    }
```

This will create a gradient of three colors, with the "pure" color **blue** positioned at exactly 30% into the gradient.

HTML   CSS   Output

```
 1
 2
 3
 4
 5
 6  .grad {
 7      backgrou
 8      width: 100
 9      height: 10
10  }
11
12
13
14
15
16
17
18
19
20
```

css   output

The `stop` value is optional. If absent, the position will be automatically set such that the color is evenly spaced between the preceding and following color-stops with positions.

For example,

```
    #el {
        background-image: linear-gradient(to right, black, blue, red)
    }
```

This will create a gradient of three colors, with the "pure" color blue positioned at exactly 50% into the gradient. By default, the start color begins at 0%, and end color at 100%.

In the next lesson, you'll get busy with gradient challenges!