

# CSS3

## LINEAR-GRADIENTS

What are  
linear-gradients?

A gradient is a graduated blend  
between **two or more colors** or  
between two tints of the same color.



In CSS3, we can use the linear-gradient value to apply **gradients** to the background of any HTML element.

Gradients are **a type of generated image**. They are not a property. This means you can use gradients wherever you have been using `url(image.png)`.

Gradients can be used as values for the **background-image** and **list-style-image** properties.

```
.one { background-image: linear-gradient(); }
```

```
.two { list-style-image: linear-gradient(); }
```

# Breaking down linear-gradients

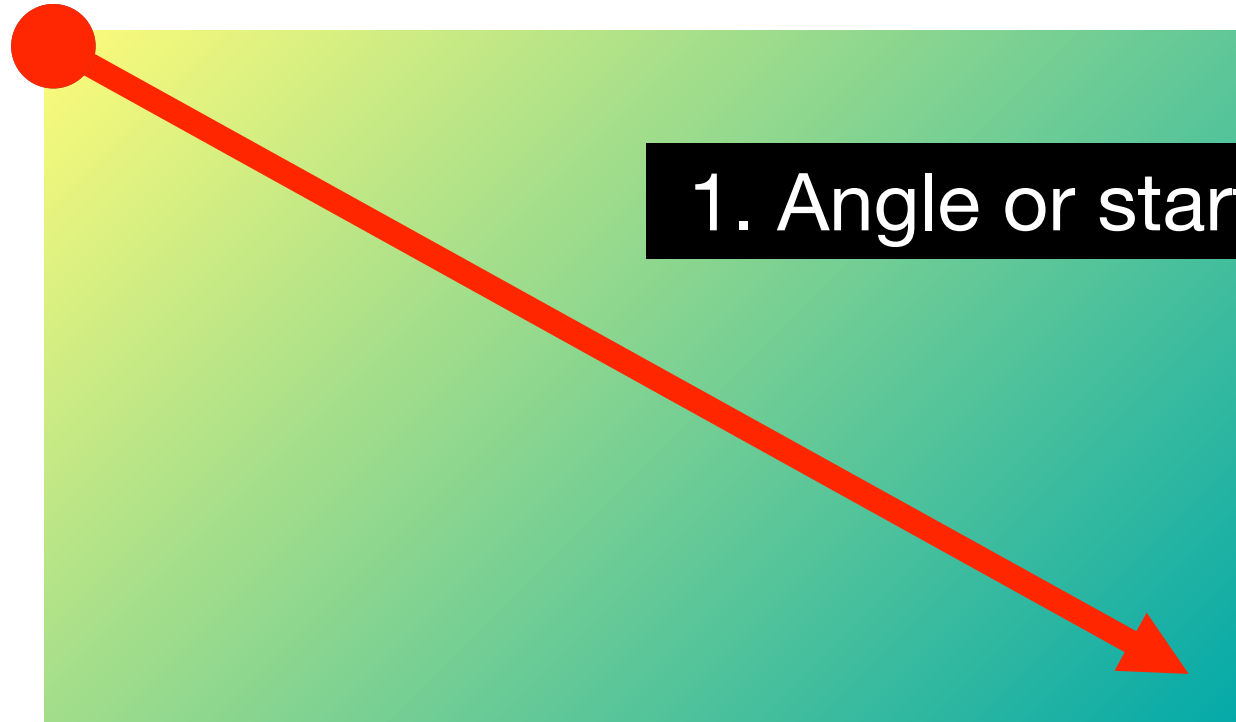


In order to create a linear-gradient, you only need **three pieces of information**:



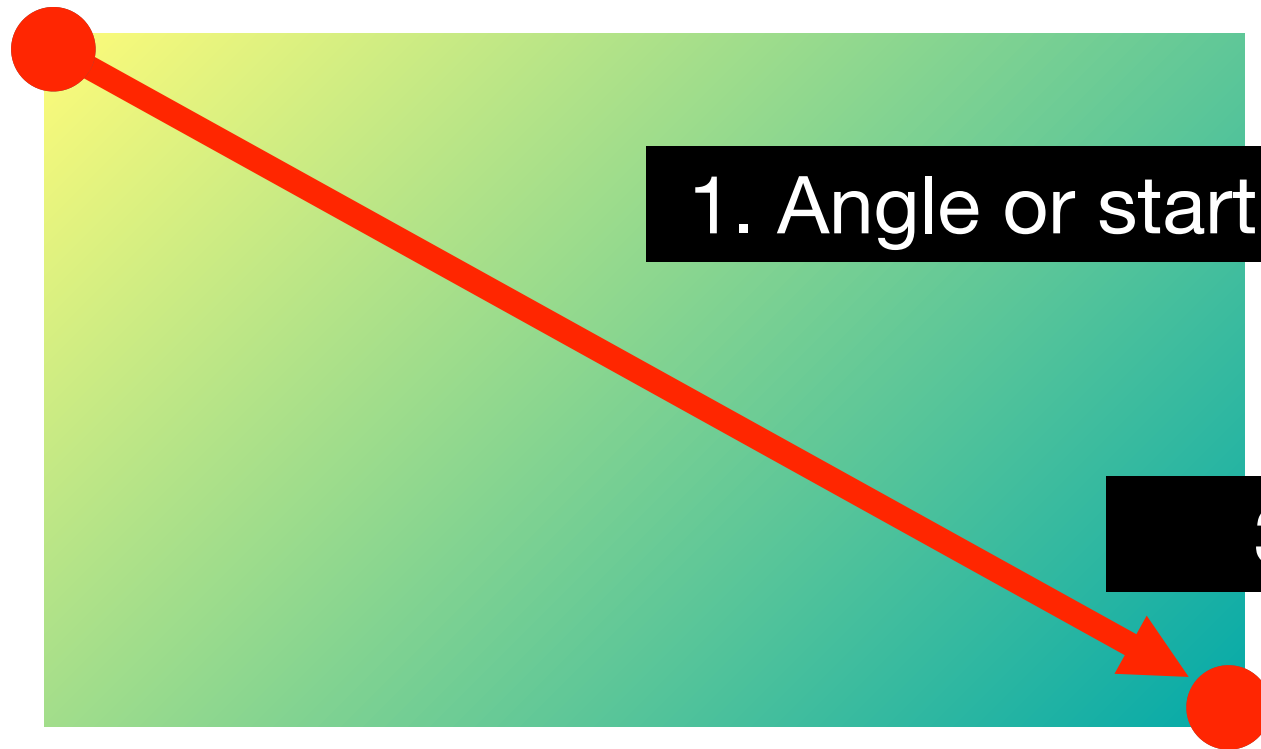
1. Angle or start point

2. Start color



1. Angle or start point

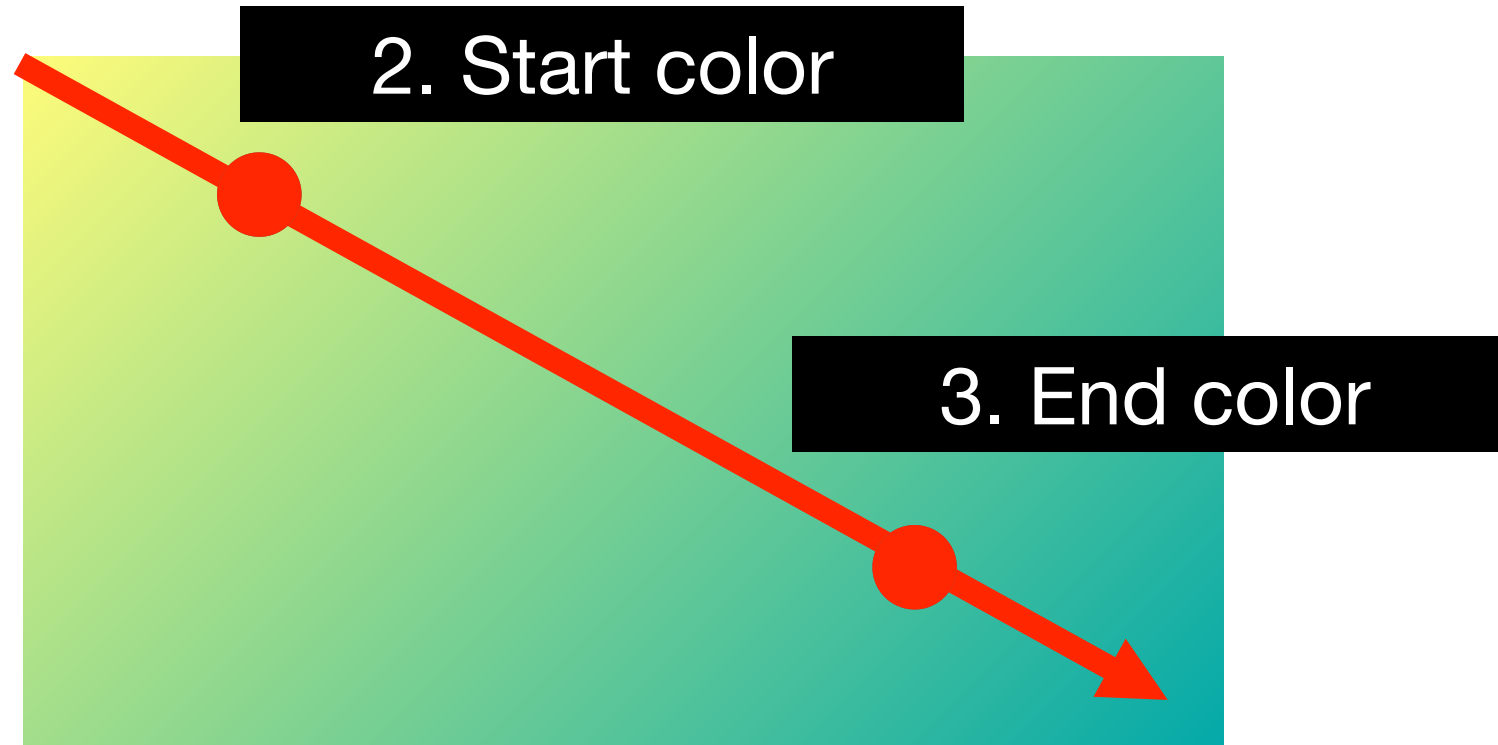
2. Start color



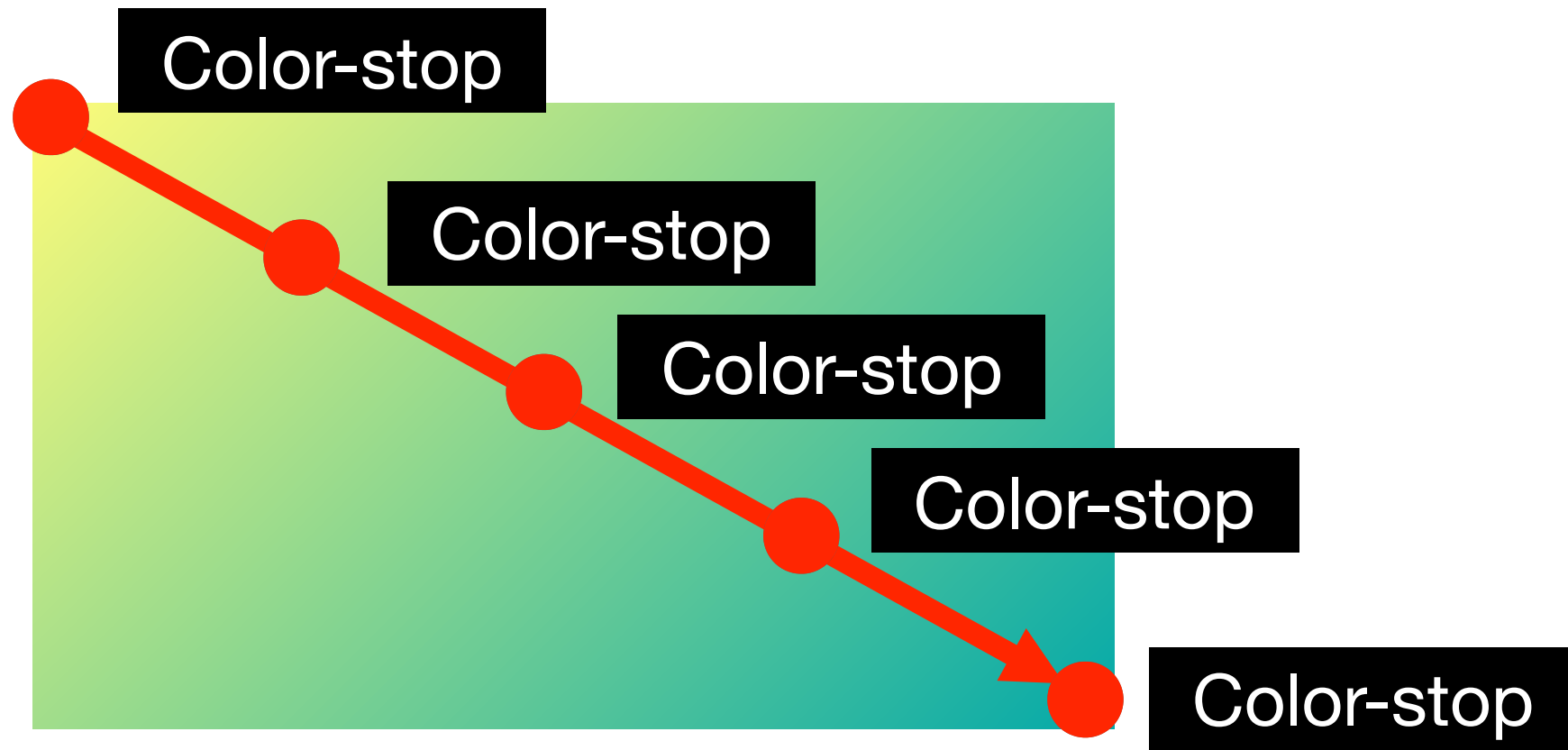
1. Angle or start point

3. End color

The start color and end color are called “**color-stops**”. These color-stops can be positioned anywhere along the gradient line.

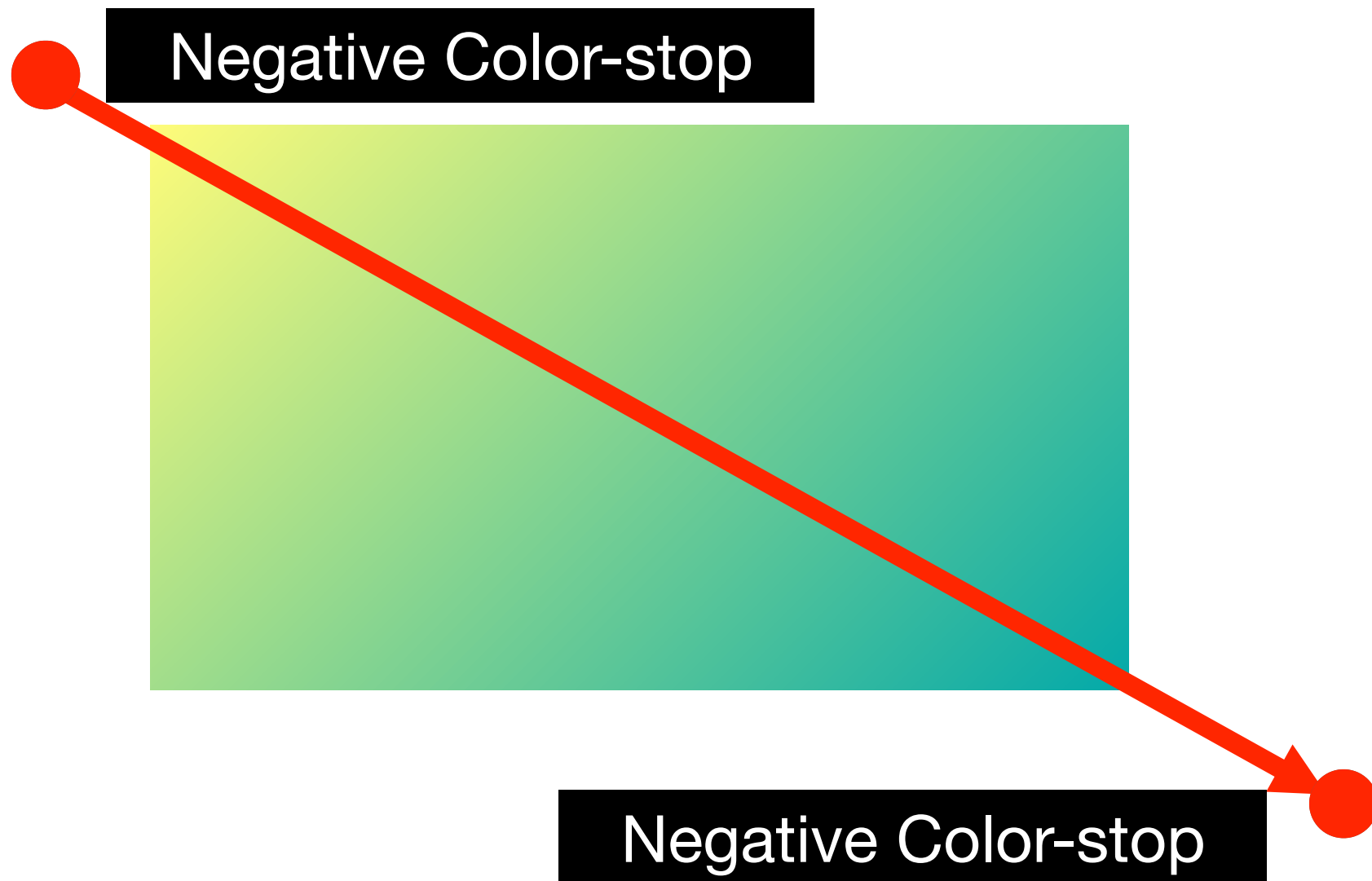


There can also be **multiple color-stops** along the gradient line.





Color-stops are allowed to have positions **before 0% or after 100%**.



Define the start  
point or angle

The <angle> and <side-or-corner> values can be defined using one of four different methods:

**leaving undefined**

**two keyword value**

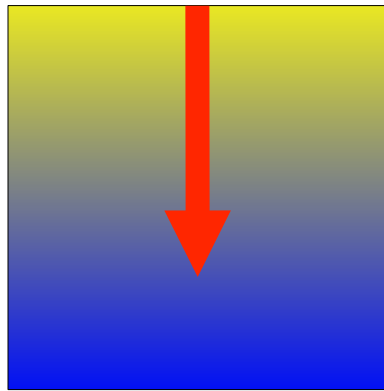
**three keyword value**

**an angle defined in degrees**

Option 1: undefined

You can leave the <angle> and <side-or-corner> values **undefined**, and the browser will use the initial value “to bottom” - which will apply the gradient from the top to the bottom of the container.

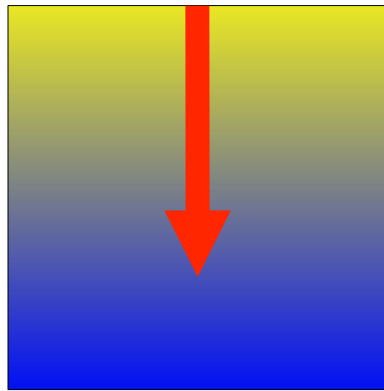
to bottom



Older webkit browsers will use the initial value “top” - which will **also apply the gradient from the top to the bottom** of the container.



top



Option 2: two  
keywords

You can define the <side-or-corner> values using a two keyword value:

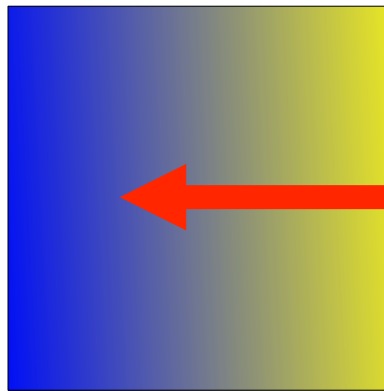
**to left**

**to right**

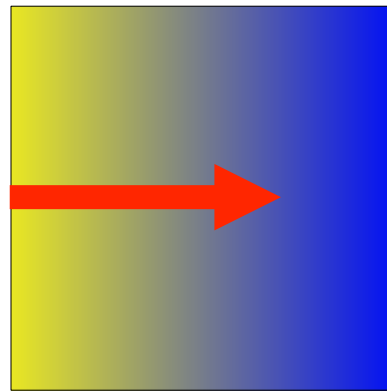
**to top**

**to bottom**

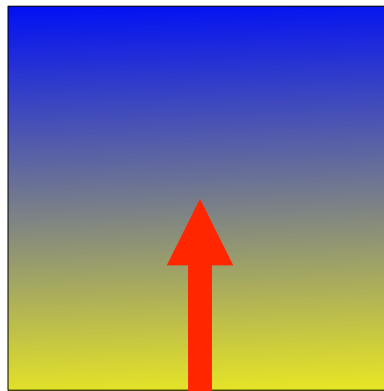
**To left** will start from the right edge of the container and spread to the left - as far as it is allowed. Older webkit browsers require “right” keyword.



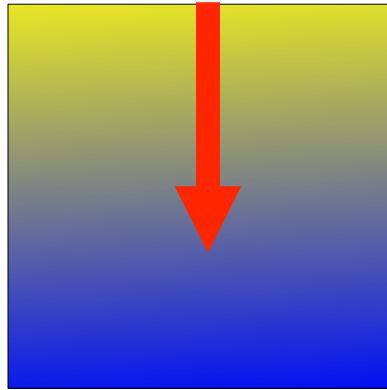
**To right** will start from the left edge of the container and spread to the right - as far as it is allowed. Older webkit browsers require “left” keyword.



**To top** will start from the bottom edge of the container and spread to the top - as far as it is allowed. Older webkit browsers require “bottom” keyword.



**To bottom** will start from the top edge of the container and spread to the bottom - as far as it is allowed. Older webkit browsers require “top” keyword.



Option 3: three  
keywords



You can define the <side-or-corner> values using a three keyword value:

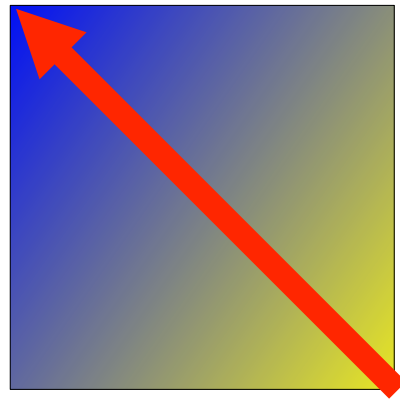
**to left top**

**to left bottom**

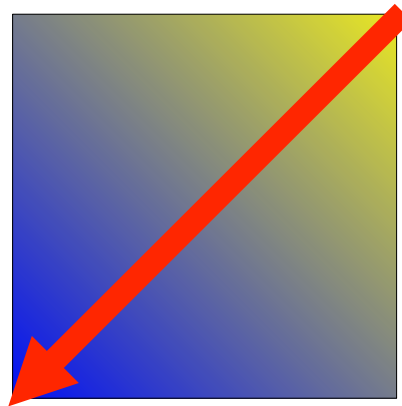
**to right top**

**to right bottom**

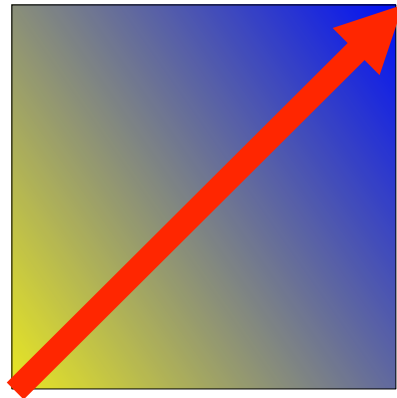
**To left top** will start from the bottom right corner of the container and spread to the top left corner - as far as it is allowed. Older webkit browsers require “bottom right” keyword.



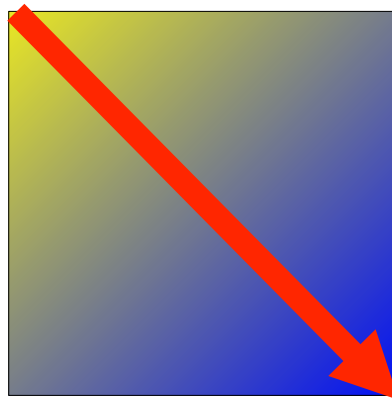
**To left bottom** will start from the top right corner of the container and spread to the bottom left corner - as far as it is allowed. Older webkit browsers require “top right” keyword.



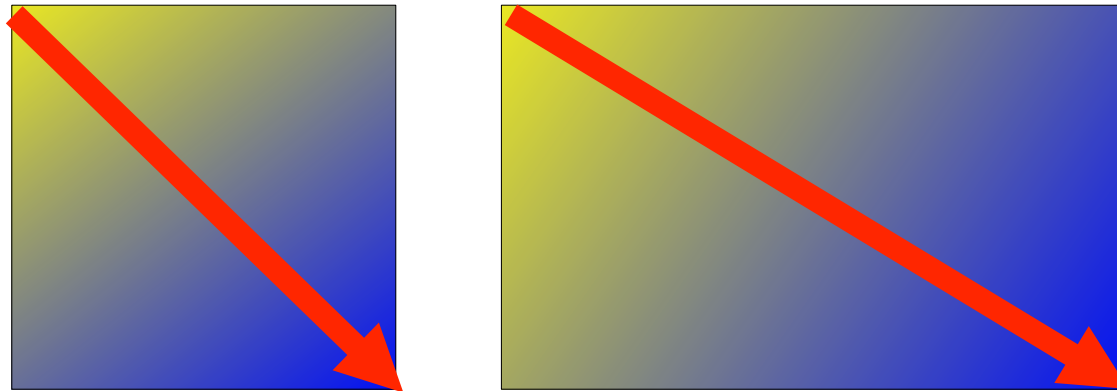
**To right top** will start from the bottom left corner of the container and spread to the top right corner - as far as it is allowed. Older webkit browsers require “bottom left” keyword.



**To right bottom** will start from the top left corner of the container and spread to the bottom right corner - as far as it is allowed. Older webkit browsers require “top left” keyword.



These gradients will spread from the **start corner to the stop corner**. The angle will depend on the box.



Option 4: angle

You can define the angle using a positive or negative degree value:

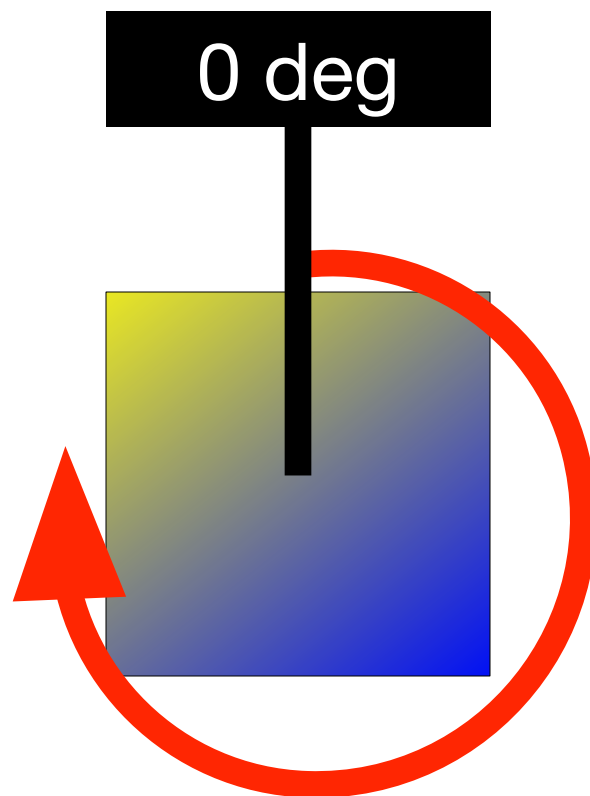
**0deg**

**90deg**

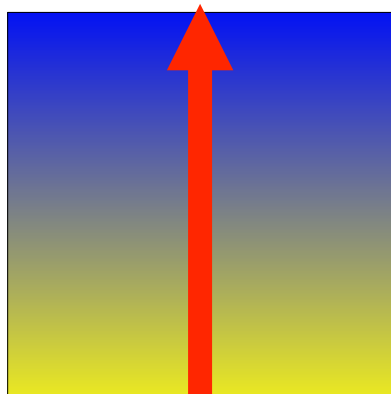
**-50deg**



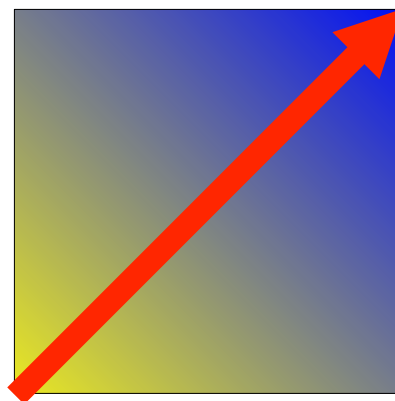
All modern browser will apply the degrees starting at the top and **travelling clockwise.**



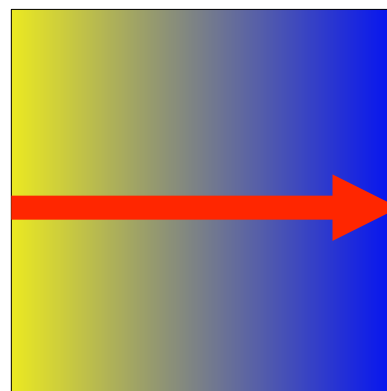
0deg



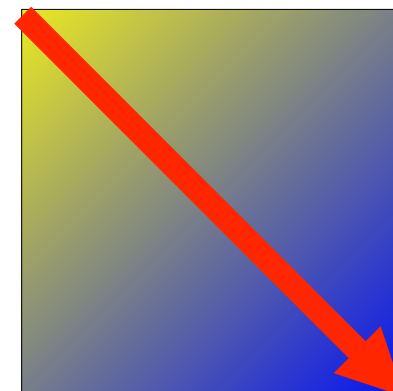
45deg



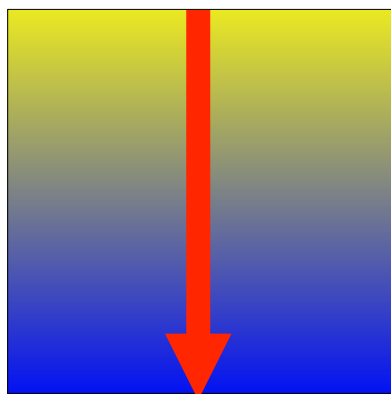
90deg



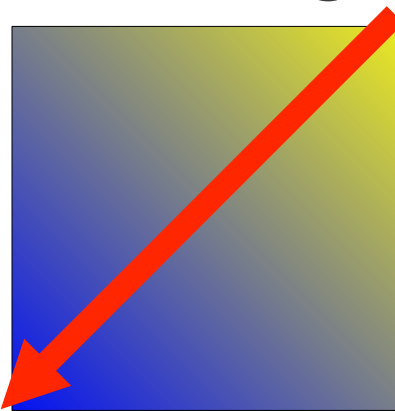
135deg



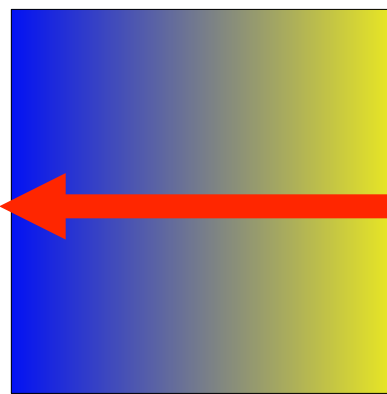
180deg



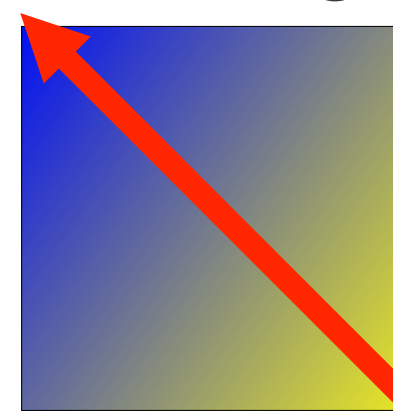
225deg



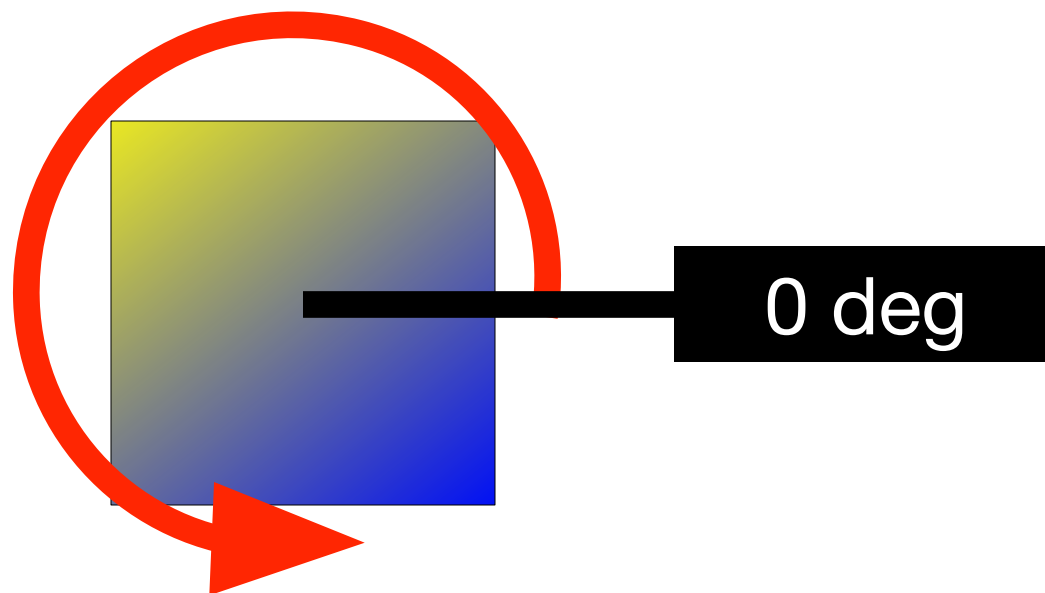
270deg



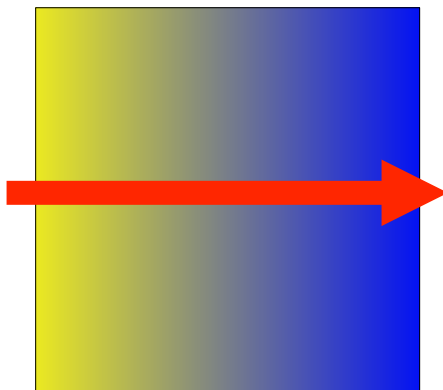
315deg



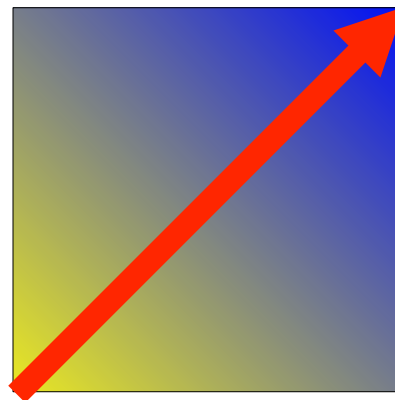
Older webkit browsers will apply the degrees starting at the right and **travelling counter-clockwise.**



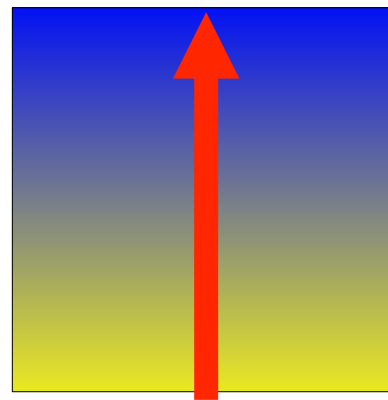
0deg



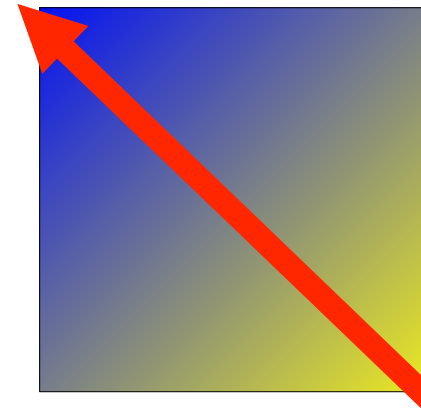
45deg



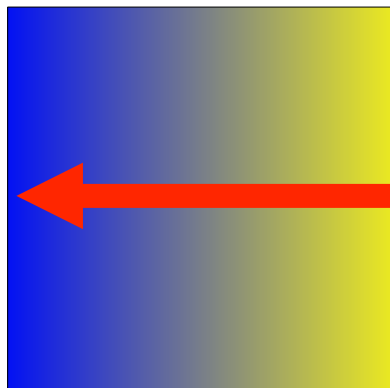
90deg



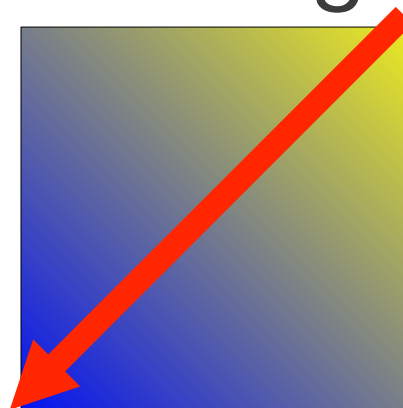
135deg



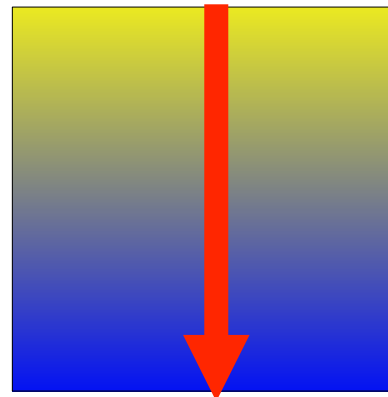
180deg



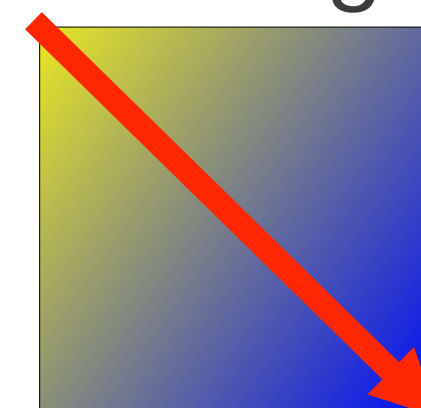
225deg



270deg



315deg



```
.test
{
    background-image:
        linear-gradient(
            to left,
        )
    ;
}
```

Define the **start**  
**color-stop**



Color-stops can be defined **using a color and an optional position**. If present, the position can be defined as a percentage or length value.

```
<color> [ <percentage> | <length> ]
```

All color-stops **require a comma after the color and optional position** -  
except for the end color-stop.

Color

Color can be specified using one of six different methods:

**keyword**

**hexadecimal notation**

**RGB/RGBA numeric notation**

**RGB/RGBA percentage notation**

**HSL notation**

**HSLA notation**

# Position

Position can be specified using a **positive or negative percentage value** such as:

0deg, 50deg or -120deg

Position can also be specified using a  
**positive or negative length value:**

em, ex, px, inch, cm, mm, pt or pc



Position can be specified using **no position value at all**. If the first color-stop does not have a position, its position is set to 0%.

```
.test
{
    background-image:
        linear-gradient(
            to left,
            yellow 5px,
        )
    ;
}
```

Define the **end**  
**color-stop**

Like the start color-stop, **the end color-stop can be defined using a color and an optional position.**

If the last color-stop does not have a position, **its position is set to 100%**. The end color-stop should not have a trailing comma.

```
.test
{
    background-image:
        linear-gradient(
            to left,
            yellow 5px,
            blue
        )
    ;
}
```

Add color-stops as  
needed

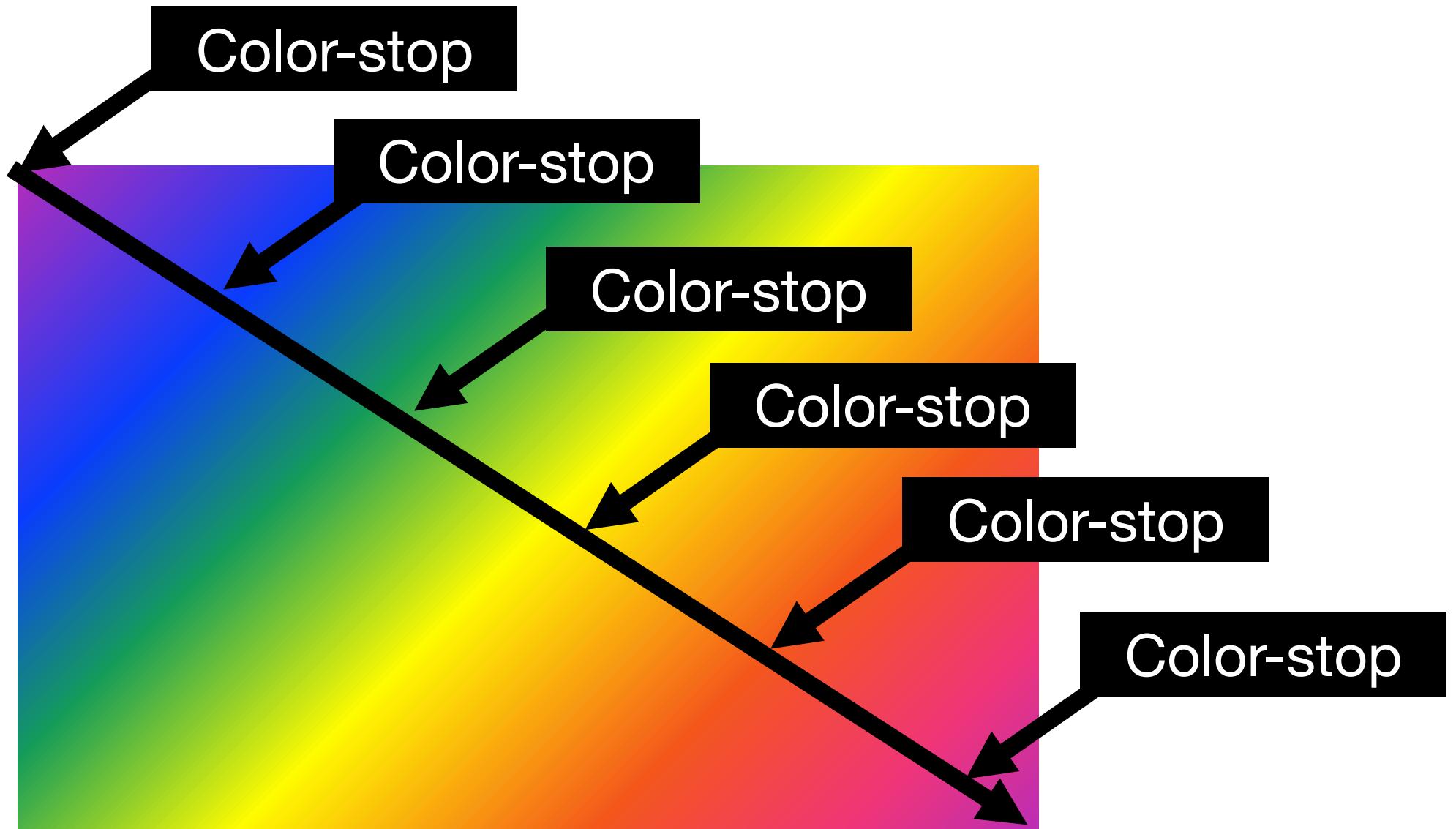
You can add **as many color-stops as you need** along a gradient path.



If no position is defined for these color stops, the browser will **equally space all color-stops** along the linear-gradient line.

```
.test
{
    background-image:
        linear-gradient(
            to left,
            yellow 5px,
            white 50%,
            blue
        )
    ;
}
```

Can you make a  
**rainbow?**



```
.test
{
    background-image:
        linear-gradient(
            to bottom right,
            violet,
            indigo,
            blue,
            green,
            yellow,
            orange,
            red);
}
```

Can you make a  
hard edged  
gradient?

Yellow color-stop



Blue color-stop

Blue color-stop

Orange color-stop

Orange color-stop

Lime color-stop

Lime color-stop

```
.test
{
    background-image:
        linear-gradient(
            yellow 50px,
            blue 50px, blue 100px,
            orange 100px, orange 150px,
            lime 150px, lime 200px
        );
}
```



Using repeating  
linear gradients

To make a repeating linear-gradient, you can change the initial value from “linear-gradient” to “**repeating-linear-gradient**”.

Start of repeating  
linear gradient

End of repeating  
linear gradient

1



More than one instance of the repeating linear-gradients will only be visible if the **initial repeating-linear-gradient finishes within the background area** of the element.

For example, if the end color-stop is set to 50%, the repeating-linear-gradient **will spread to 50% of the way across the container**, and then repeat again.

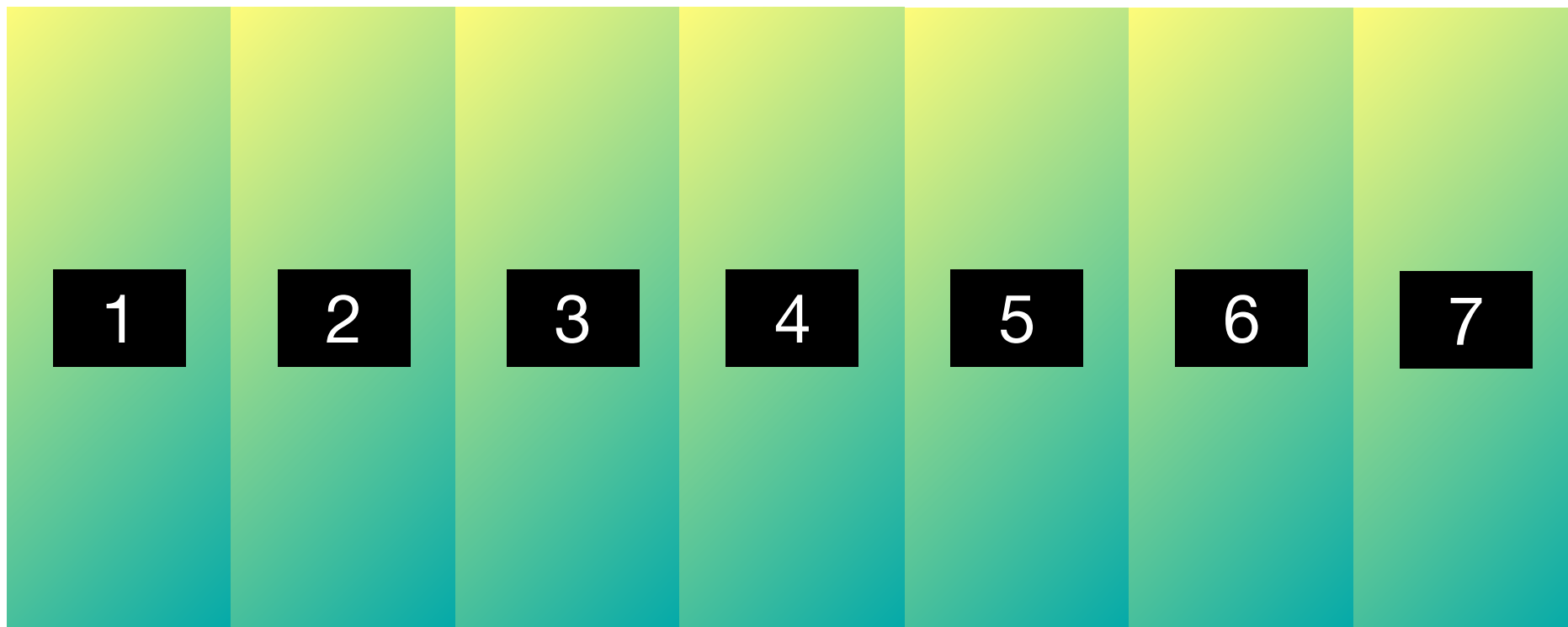
```
.test
{
    background-image:
        repeating-linear-gradient(
            to left,
            yellow,
            blue 50%
        )
    ;
}
```



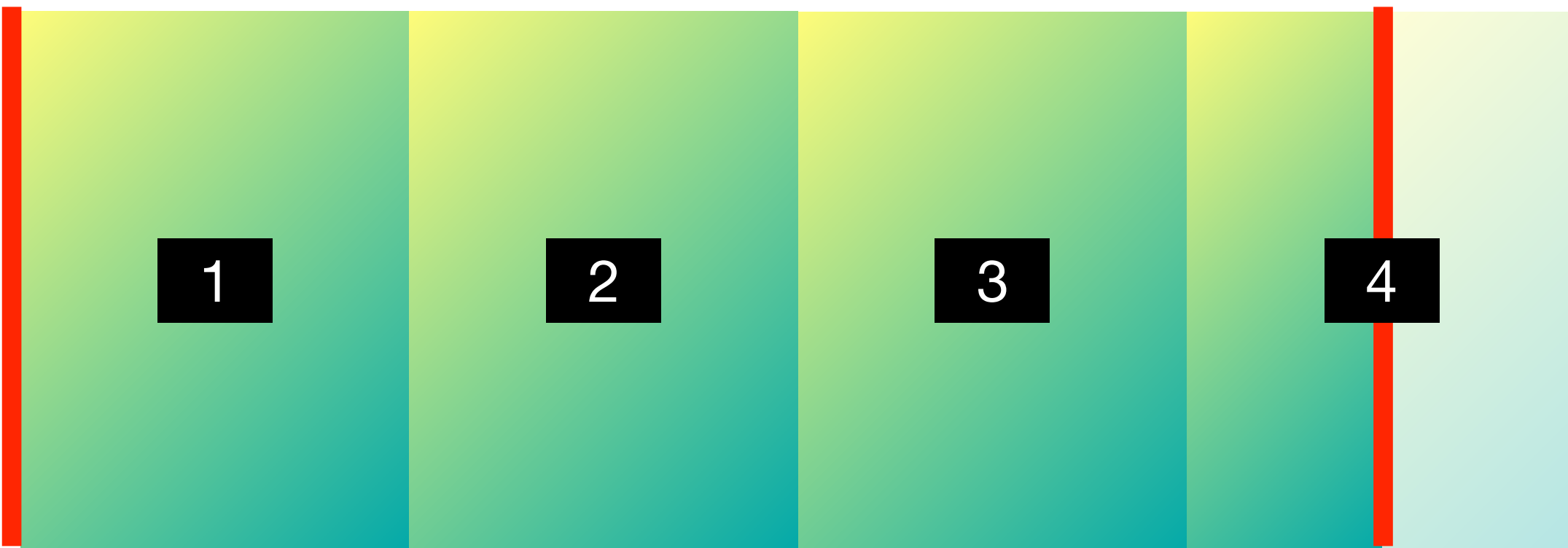
If the end color-stop is set to a pixel value of 20px, the repeating-linear-gradient **will spread to 20px**, and then repeat as often as possible across the container.



```
.test
{
    background-image:
        repeating-linear-gradient(
            to left,
            yellow,
            blue 20px
        )
    ;
}
```



The last repeating-linear-gradient **may be cut off** if it does not fit within the width of the container.



Using multiple  
linear-gradients

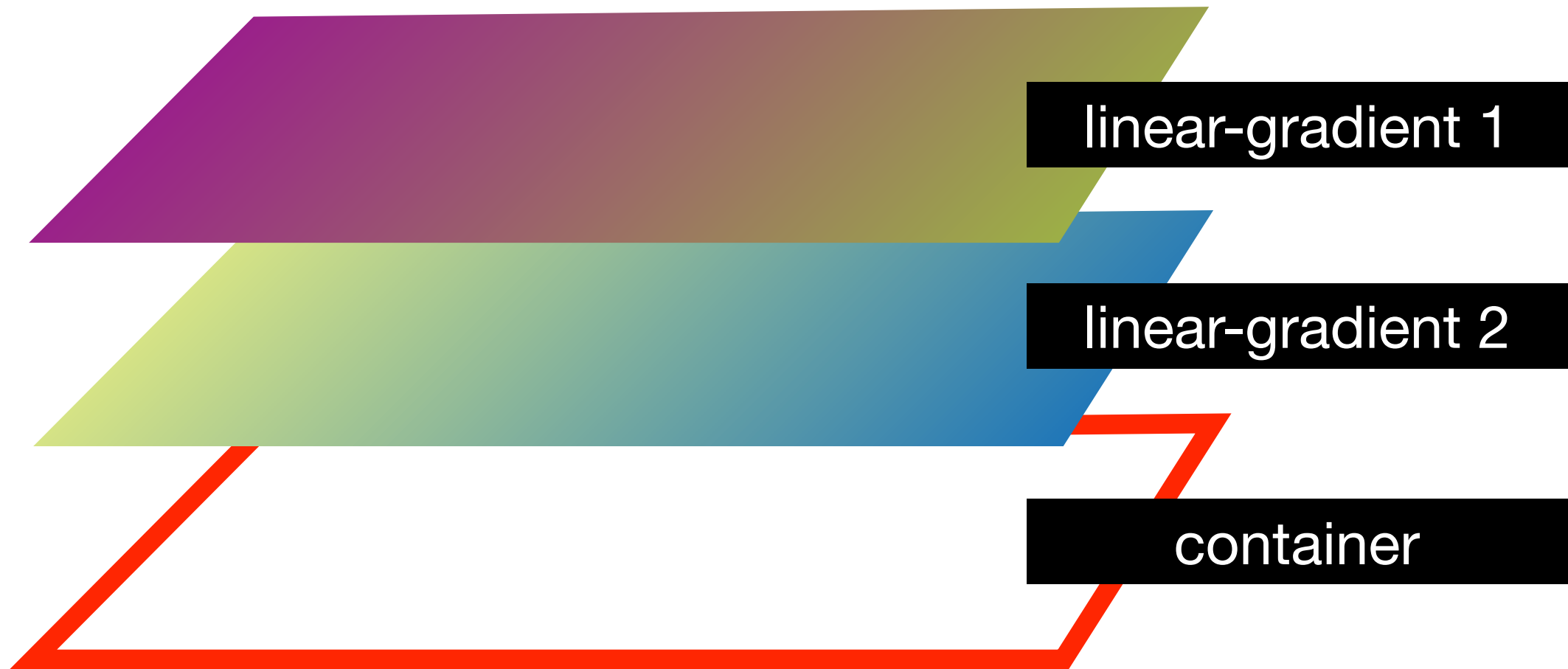
We can apply **multiple linear-gradients** within the one container. Each linear-gradient must be separated by a comma. There must be a semi-colon and no comma after the last linear-gradient.

```
.example4
{
    background-image:
        linear-gradient(purple, lime),
        linear-gradient(lime, blue);
}
```

Linear-gradients that are defined first are **applied on top of later linear gradients**.



Linear-gradients cover the entire background area of the element, so the **top linear-gradient will cover any other linear-gradients below.**



We can prevent linear-gradients from completely overlapping other linear-gradients below by **setting background-size, background-position and background-repeat** on each individual linear-gradient.

The background-size, background-position and background-repeat values can be set as **comma-separated values**. The order of the values will match the linear-gradient order.

```
.example4
{
    background-image:
        linear-gradient(purple, lime),
        linear-gradient(lime, blue);
    background-size:
        100px 100px, 80% 80%;
    background-repeat:
        no-repeat, no-repeat;
    background-position:
        0 0, right bottom;
}
```

```
.example4
{
    background-image:
        linear-gradient(purple, lime),
        linear-gradient(lime, blue);
    background-size:
        100px 100px, 80% 80%;
    background-repeat:
        no-repeat, no-repeat;
    background-position:
        0 0, right bottom;
}
```

# Browser support

# CSS Gradients - CR

Global90.06% + 0.3% = 90.36%

unprefixed:79.48%

Method of defining a linear or radial color gradient as a CSS image.

Current alignedUsage relativeShow all

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
								4.1	
8			43					4.3	
9		40	44					4.4	
10		41	45	8		8.4		4.4.4	
11	12	42	46	9	32	9.1	8	44	46
	13	43	47		33				
		44	48		34				
		45	49						



A word on fallbacks

As gradients are not supported by all browsers, you may wish to include one or **two different types of fallback.**

A **background-color** can be used if you would like the linear-gradient to fail to a flat color.

```
.test
{
    background-color: yellow;
    background-image:
        linear-gradient(yellow,blue);
}
```

A **background-image** can be used if you want the gradient to fail to a representative gradient.

```
.test
{
    background-color: yellow;
    background-image: url(tint.jpg)
    background-image:
    linear-gradient(yellow,blue);
}
```

Modern browsers **should ignore the background-image** and use the gradient only.



# Russ Weakley

Max Design

**Site:** [maxdesign.com.au](http://maxdesign.com.au)

**Twitter:** [twitter.com/russmaxdesign](https://twitter.com/russmaxdesign)

**Slideshare:** [slideshare.net/maxdesign](https://slideshare.net/maxdesign)

**Linkedin:** [linkedin.com/in/russweakley](https://linkedin.com/in/russweakley)