**Linear-gradient详解 教程**

想要在你的网站有一个好看的渐变背景吗？background-image:linear-gradient(red,blue)，就能很轻易的实现一个red至blue的渐变背景图片。就是这么的简单。

感觉老话重提，有点无聊。如果您想了解更多这方面的知识，建议你阅读以前整理过的[相关文章](https://www.w3cplus.com/blog/tags/41.html)。我也推荐[css-tricks](https://css-tricks.com/css3-gradients/)和[MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/linear-gradient)上的相关文章。

还在这里吗？好了，让我们看一些linear-gradient实际上工作的一些细节，从而更好的理解它是如何工作的。首先了解linear-gradient的语法：

linear-gradient([<angle> | to <side-or-corner>]? , <color-stop-list>)

这个函数（特性）接受的第一个参数是渐变的角度，他可以接受一个表示角度的值（可用的单位deg、rad、grad或turn）或者是表示方向的关键词（top、right、bottom、left、left top、top right、bottom right或者left bottom）。第二个参数是接受一系列颜色节点（终止点的颜色）。

**渐变容器（渐变框）**

一个渐变图像和传统的背景图像不一样，它是没有维度（尺寸限制），它是无限的。那么决定渐变图像可见区域是由渐变容器大小来决定的。

通常，如果给一个DOM元素的background-image使用linear-gradient，那么其（渐变）显示区域就是元素的border-box区域（如果不了解元素的border-box区域，建议先阅读box-sizing相关的文档）。其实也是background-color或者说通过url引入背景图像的显示区域。

然而，如果你通过CSS的background-size设置一个尺寸，比如说200px \* 200px，这个时候渐变容器（渐变尺寸）就是background-size设置的大小200px \* 200px。在没有使用background-position设置为其他值时，它默认是显示在DOM元素的左上角（也就是background-position: left top）。

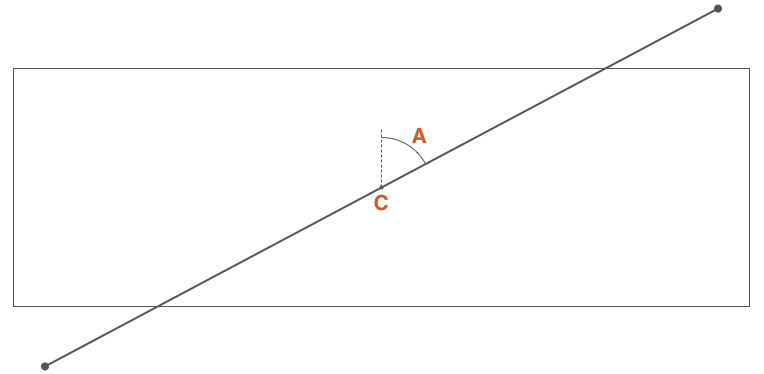
在CSS中渐变就是background的background-image，也就是说，适用于背景图像的CSS属性都适合于渐变。

**渐变线**

在渐变容器中，穿过容器中心点和颜色停止点连接在一起的线称为渐变线。在下节介绍渐变角度相关的知识时，能帮助你更好的理解渐变线，所以更多的细节我们在下一节中介绍。

**渐变角度**

很明显，使用linear-gradient是通过渐变的角度来控制渐变的方向。接下来我们一起来了解其中更多的细节。

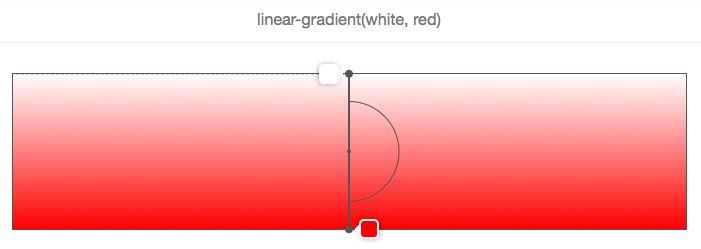


C点渐变容器中心点，A是通过C点垂直线与通过C点渐变线的夹角，这个角称为渐变角度。

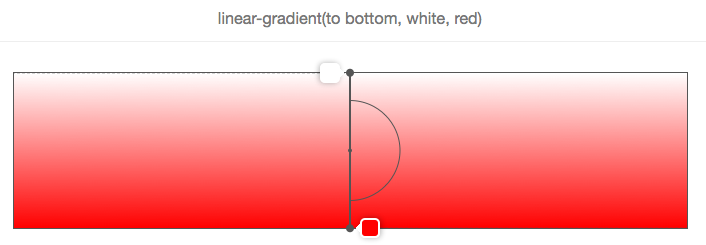
可以通过下面两种方法来定义这个角度：

* 使用关键词：to top、to bottom、to left、to right、to top right、to top left、to bottom right和to bottom left
* 使用带单位数字定义角度，比如45deg、1turn等

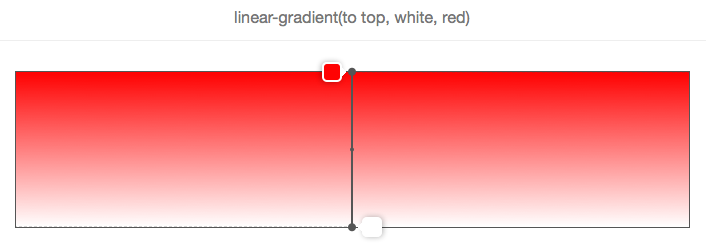
如果省略角度值的设置，那默认是to bottom（对应180deg或者.5turn）：

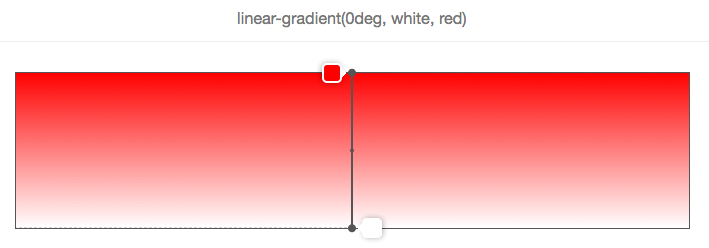


在上面的示例中，渐变角度是没有设置，white至red渐变色从top至bottom渐变，它和使用to bottom关键词得到的效果是一样的，如下所示：



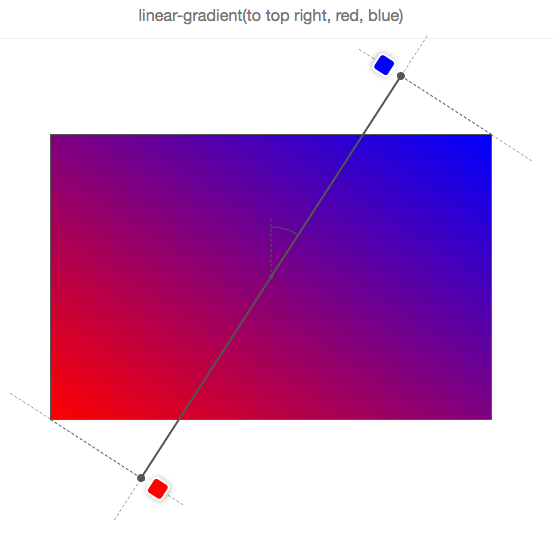
下面两张图的效果是使用to top和0deg，它们的效果也是一样的：





另一个是使用顶角关键词重要的一点是它依赖于渐变容器的尺寸，比如to top right（或者其它顶角关键词）。

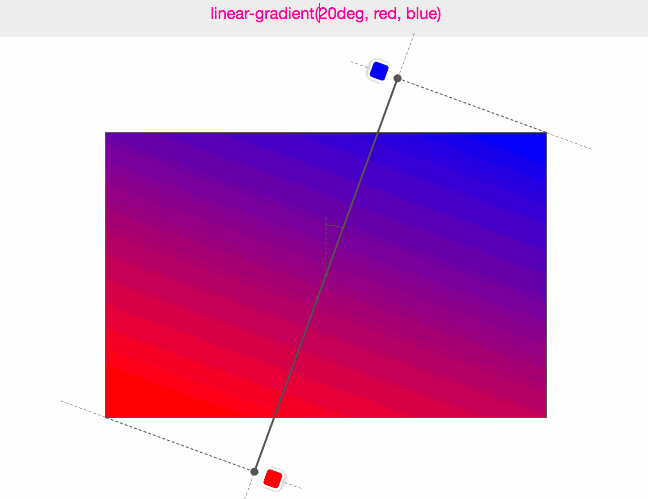
如果你想要一个red至blue的渐变，方向是至元素的top right。逻辑上，blue应该在元素的右上角，以及中间的紫色渐变周围应该形成一条直线，从左上角至右下角穿过。如下图所示：



所以to top right并不意味着渐变线穿过右上角，也就是说渐变角度并不意味着是45deg。

也就是说，如果linear-gradient使用顶角关键词时（to top right、to top left、to bottom right和to bottom left），渐变线首先通过元素中心点并且与顶点垂直相交，与中心点垂直线构成的夹角才是渐变角度。

让我们看看渐变角度动态变化时，渐变线是怎么移动的：

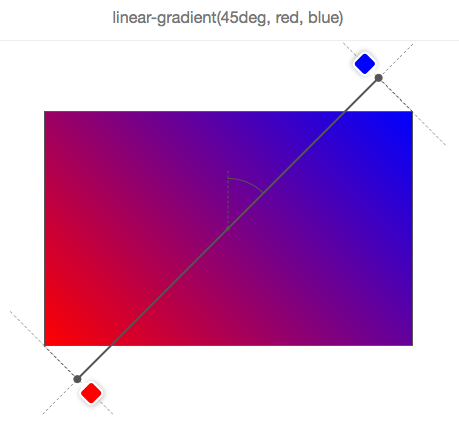


回顾一下渐变角度：

* 角度是渐变线与渐变容器中心点向上垂直线之间的夹角
* 0deg的意思就是to top
* 角度的默认值（也就是角度没有设置），它的值是to bottom，也和180deg相同
* 顶角关键词和渐变容器尺寸有关

**渐变线长度a**

之前我们看到渐变色停止分布沿着渐变线是需要解释的一件事情。你可能已经注意到了，在前面的示例中，停止的渐变颜色有时候在渐变容器以外的位置，这看起来有点奇怪，但如果你知道其中的逻辑之后，你就不会这么认为了。先看一下这个示例：



我们想要一个red至blue的渐变，渐变的角度是45deg，因为渐变容器的比例，渐变线不能通过右上角。但浏览器想要做什么（规范告诉它做什么），能使右上角是blue。

如果我们让渐变线的开始和结束都在渐变容器的边缘，那么blue将会覆盖渐变容器更大的区域，渐变不会有更多的扩散。

因此，为了做到这一点，渐变线有时不得不延长到渐变容器之外。其实很容器知道它的开始和结束位置。通过最近的角落画一条垂直于渐变线的线，与渐变线交叉的地方，就是渐变的开始和结束位置（[规范中做出了很好的解释](https://dev.w3.org/csswg/css-images/#linear-gradient-syntax)）。

事实上，如果W是渐变容器的宽度，H是渐变容器的高度，A是渐变角度，那么渐变线的长度可以通过下面的公式计算：

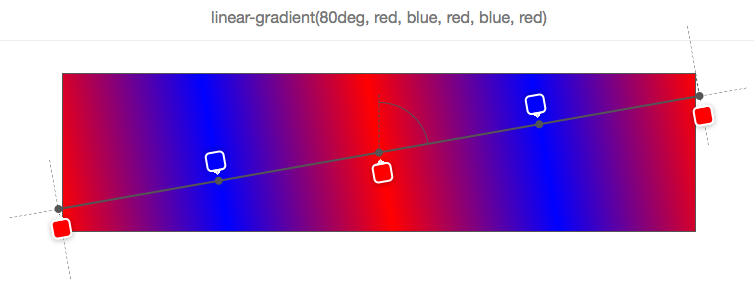
abs(W \* sin(A)) + abs(H \* cos(A))

**渐变色节点（Color stops）**

渐变色的每一个可以这样定义：

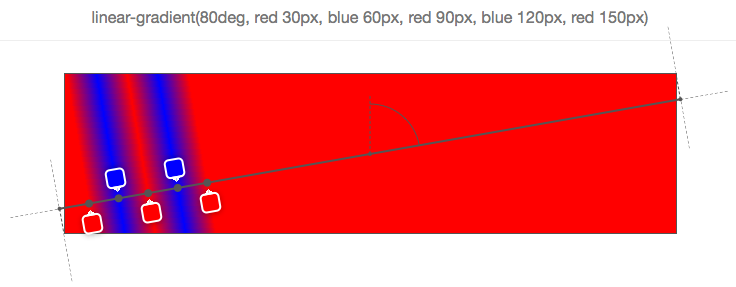
<color> [<percentage> | <length>]?

因此不是强制性来指定颜色在渐变线的位置。例如：



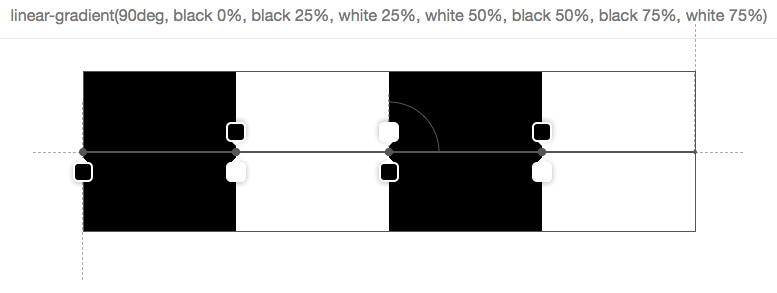
如果没有显式指定颜色在渐变线上的位置，这将交给浏览器来确定颜色在渐变线上的位置。最简单的情况下只有两个颜色，颜色1将被放置在渐变线0%位置（渐变线开始位置），颜色2将被放置在100%位置处（渐变线的结束点）。如果有三个颜色，那么颜色1在渐变线的0%，颜色2在渐变线的50%，颜色3在渐变线的100%。在上面的这个示例中，有五个颜色，那么它们的位置分别在0%、25%、50%、75%和100%。它们将沿着渐变线平均分布渐变颜色。

当然，也可以在渐变线上显式自定义渐变颜色在渐变线的位置。每个位置可以用百分比表示（相对于渐变线计算），也可以是任何一个CSS长度单位。比如下面这个示例：



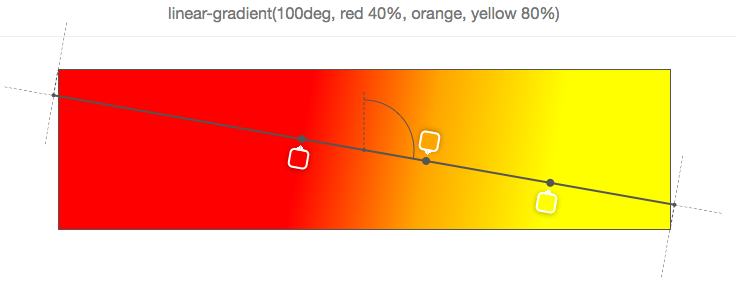
正如你所看到的，五个颜色的每个颜色都有自己的位置，而且是以像素为单位。这些位置从渐变线的开始位置处开始计算。

使用这些位置，你可以想出各种各样的漂亮效果。这样你可以做一个多色渐变：



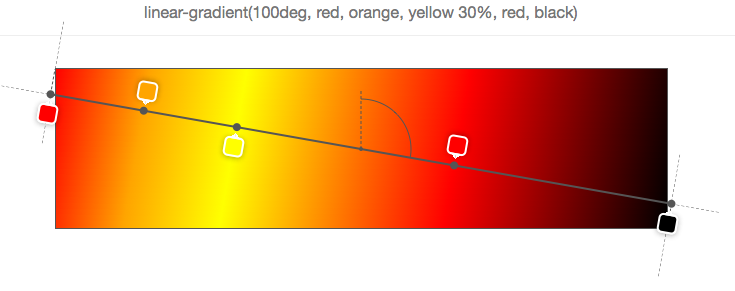
上图中，有七个颜色，其中下一个颜色是在上一个颜色开始位置，这意味弟浏览器不需要填满两个颜色之余间的空间。

当然这样蛮好的也很有趣，如果你把颜色位置配合一起来使用会是什么样的情形。然后让浏览器自动分配你省略的颜色位置。

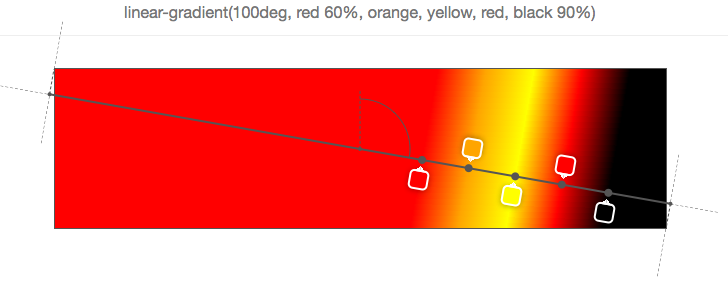


在上面的示例中，第二个颜色orange没有明确的指定其在渐变线上的位置，所以浏览器会自动计算出其位置。它可以根据第一个位置和下一个位置很容易计算出来。但如果有多个颜色没有指定位置，或者前一个或后一个都没有指定位置，那它就变得越来越复杂。

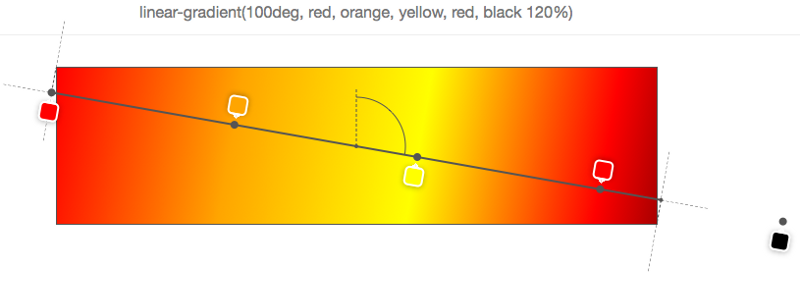
看下面这个示例：



在上图中，只有第三个颜色yellow指定了位置，在渐变线的30%处。为了很好的分发，它把第一个颜色red放置在渐变线的0%处，最后一个颜色black放置在渐变线的100%处。第二个颜色orange放置在渐变线0%至30%的中间位置，第四个颜色red放置在渐变线30%至100%中间位置。

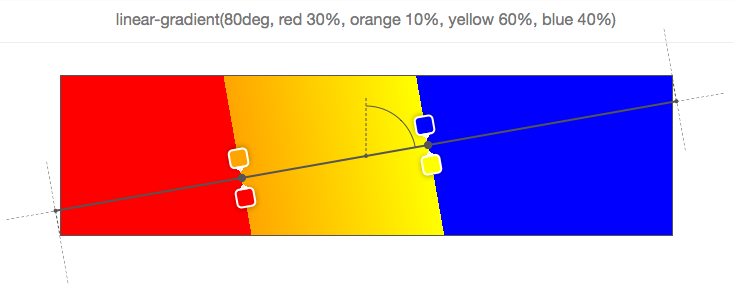


上图第一个和最后一个颜色放置在渐变线指定位置，剩下的颜色平均分布在两者之间。

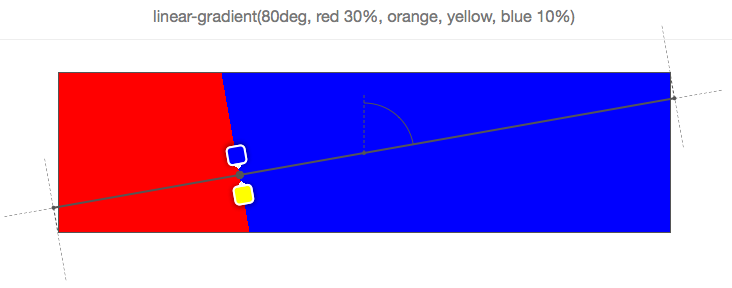


当然，如果是0%和100%之间，我们很容易控制。但也有会超出这个范围。比如上面的示例，最后一个颜色是在渐变线的120%位置处，因此其他颜色也将根据这个位置平均分布（默认的起始位置仍然是0%，在这个示例中）。

如果你想让你的浏览器工作更多，为什么不能按顺序指定颜色在渐变线上的位置呢？事实上，颜色点位置是按照你预计的指令操作，并不会阻止你不按其位置顺序来操作。但如果后面的值比前面的值更小时，浏览器会自动做相应的纠正处理。比如：



让我们从第一个颜色red开始，其定位在渐变线的30%位置处，第二个颜色orange在10%位置，但这是错误的，正如上面所说的，颜色的停止点是一个增量。这个时候，浏览器将会纠正第二个颜色的位置，它将会和前一个颜色的位置一样，也分布在渐变线的30%位置。然后第三个颜色yellow分布在渐变线的60%位置处，但紧随其后的第四个颜色blue为40%，浏览器同样会纠正并设置其位置与前一个颜色位置相同。



最后一点，在上面这个例子中，最后一个颜色blue是不正确的位置，因此浏览器将会纠正它的位置与之前的位置相同，在这种情况之下并不是与其相邻的颜色yellow，也不会是orange，它会追溯到第一个颜色red位置处。因此，red和blue都分布在渐变线的30%处，因此其中yellow和orange两颜色都将不可见。

**工具**

文章中的截图都是从我自己在[Codepen写的一个简单工具](https://codepen.io/captainbrosset/pen/ByqRMB)获取的，你可以在输入框中输入任何一个渐变的值，你可以看到渐变效果以及渐变线，渐变角度和渐变颜色的位置。

目前这个工具还存在各种各样的缺陷和局限性（见JavaScript中的注释），所以不要有过高的期望，当然你也可以在这个基础上完善这个工具，帮助大家更好的理解渐线渐变。

著作权归作者所有。  
商业转载请联系作者获得授权,非商业转载请注明出处。  
原文: <https://www.w3cplus.com/css3/do-you-really-understand-css-linear-gradients.html> © [w3cplus.com](https://www.w3cplus.com/)