<meta charset="gbk"/>

中文的编码格式: gbk, gb2312, utf-8, big5

英文的编码格式:iso8859-1,…

GBK: 汉字国标扩展码,基本上采用了原来GB2312-80所有的汉字及码位，并涵盖了原Unicode中所有的汉字20902，总共收录了883个符号， 21003个汉字及提供了1894个造字码位。 Microsoft简体版中文Windows 95就是以GBK为内码，又由于GBK同时也涵盖了Unicode所有CJK汉字，所以也可以和Unicode做一一对应。  
  
GB码，全称是GB2312-80《信息交换用汉字编码字符集 基本集》，1980年发布，是中文信息处理的国家标准，在大陆及海外使用简体中文的地区（如新加坡等）是强制使用的唯一中文编码。P-Windows3.2和苹果OS就是以GB2312为基本汉字编码， Windows 95/98则以GBK为基本汉字编码、但兼容支持GB2312。GB码共收录6763个简体汉字、682个符号，其中汉字部分：一级字3755，以拼音排序，二级字3008，以偏旁排序。该标准的制定和应用为规范、推动中文信息化进程起了很大作用。  
  
GBK编码是中国大陆制订的、等同于UCS的新的中文编码扩展国家标准。GBK工作小组于1995年10月，同年12月完成GBK规范。该编码标准兼容GB2312，共收录汉字21003个、符号883个，并提供1894个造字码位，简、繁体字融于一库。

GB2312(1980年)一共收录了7445个字符，包括6763个汉字和682个其它符号。汉字区的内码范围高字节从B0-F7，低字节从A1-FE，占用的码位是72\*94=6768。其中有5个空位是D7FA-D7FE。  
  
GB2312支持的汉字太少。1995年的汉字扩展规范GBK1.0收录了21886个符号，它分为汉字区和图形符号区。汉字区包括21003个字符。2000年的GB18030是取代GBK1.0的正式国家标准。该标准收录了27484个汉字，同时还收录了藏文、蒙文、维吾尔文等主要的少数民族文字。现在的PC平台必须支持GB18030，对嵌入式产品暂不作要求。所以手机、MP3一般只支持GB2312。

E:\中青\图像\image\hetao.jpg 绝对路径

..\image\hetao.jpg 相对路径

<form action="#" method="post">

<p><input type="button" name="按钮" value="普通按钮" /></p>

<p><input type="checkbox" name="复选框" checked="checked" /></p>

<p><input type="color" name="设置颜色的对话框" /></p>

<p><input type="date" name="日期" /></p>

<p><input type="datetime" name="日期时间" /></p>

<p><input type="datetime-local" name="本地日期时间" /></p>

<p><input type="email" name="邮箱" /></p>

<p><input type="file" name="文件域" /></p>

<p><input type="hidden" name="隐藏域" /></p>

<p><input type="image" name="图片按钮" src="img/but.png" /></p>

<p><input type="month" name="年月" /></p>

<p><input type="number" name="数字框，只能输入数字" /></p>

<p><input type="password" name="密码框" /></p>

<p><input type="radio" name="单选按钮" checked="checked" /></p>

<p><input type="range" name="范围选择" /></p>

<p><input type="reset" name="重置按钮" value="重置" /></p>

<p><input type="search" name="搜索框" /></p>

<p><input type="submit" name="提交按钮" value="提交" /></p>

<p><input type="tel" name="电话" /></p>

<p><input type="text" name="文本框" /></p>

<p><input type="time" name="时间" /></p>

<p><input type="url" name="网址" /></p>

<p><input type="week" name="星期" /></p>

<p>下拉列表<select>

<option value="0">请选择</option>

<option value="1">星期一</option>

<option value="2">星期二</option>

<option value="3">星期三</option>

</select>

</p>

<p>

<textarea readonly="readonly">多行文本域</textarea>

</p>

<p>字段集

<fieldset>

<legend>分组框</legend>

</fieldset>

</p>

</form>

同一组单选按钮的name的值要相同

同一组复选框的name的值尽量相同



可能从字面意思上可以看出这段的意义

这应该算是前端的一种优化方式

DNS主机名解析有时会出现拖慢网站速度的问题。现代浏览器当遇到DNS解析时已经十分聪明——用户在跟随某个链接之前，浏览器先尝试解析域名再将其进行缓存。  
DNS Prefetching特性允许开发者手动控制，告诉浏览器需要解析哪个域名

# [CSS里面的长度单位px/pt/em/in/pc/mm/cm解释](http://www.cnblogs.com/913815ccmm/p/4858638.html)

在CSS中，度量单位分为两种：  
  
相对单位：px/em/ex/%  
绝对单位：cm/pt/in/pc/mm  
  
这些单位的具体解释  
  
px  
相对长度单位。像素（Pixel）。  
像素是相对于显示器屏幕分辨率而言的。譬如，WONDOWS的用户所使用的分辨率一般是96像素/英寸。而MAC的用户所使用的分辨率一般是72像素/英寸。  
  
em  
相对长度单位。相对于当前对象内文本的字体尺寸。  
如当前行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。  
  
ex  
相对长度单位。相对于字符“x”的高度。此高度通常为字体尺寸的一半。  
如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。  
  
pt  
绝对长度单位。点（Point）。  
1in = 2.54cm = 25.4 mm = 72pt = 6pc  
pc  
绝对长度单位。派卡（Pica）。相当于我国新四号铅字的尺寸。  
1in = 2.54cm = 25.4 mm = 72pt = 6pc  
in  
绝对长度单位。英寸（Inch）。  
1in = 2.54cm = 25.4 mm = 72pt = 6pc  
mm  
绝对长度单位。毫米（Millimeter）。  
1in = 2.54cm = 25.4 mm = 72pt = 6pc  
cm  
绝对长度单位。厘米（Centimeter）。  
1in = 2.54cm = 25.4 mm = 72pt = 6pc  
究竟有什么区别呢？  
    首先要明白一点，那就是屏幕分辨率。我们知道，在不同的分辨率下，像素点的大小是不同的。所以同一个网页，以px作长度单位时，在不同的分辨率下显示的大小是不同的。在低分辨率下，像素点较大，细节不够清晰，显示的页面也大，但模糊不清。  
  
    实际上,所有的单位,无论是相对还是绝对单位,最终都是转化为px为单位的---在屏幕上的显示。所以一般来说，在网页制作时，基本单位都选择px而不是pt，因为pt也是通过浏览器的DPI转换成px显示（比如FireFox的DPI是96，则有9pt = 12px）。不仅pt，cm/in/mm等都是转换成px的。所以无论用绝对还是相对，在不同分辨率下都是会变的。不要以为我把长度设置为3cm，它就会在不同的分辨率下保持3cm不变。  
  
那又为什么出现绝对单位和相对单位呢?  
  
    我觉得pt还是很有用的一个单位，在打印时尤其如此。现在有的网页已经实现了显示是一个页面，打印用另一种方式，我想在打印上采用的就是pt吧，因为针对不同的分辨率，打印出来的页面都是一样的大小。这就是绝对单位的第一个特征。但是依然要记住，在显示上，这个单位代表的不是真实的物理长度，它也是要根据像素的实际大小（这个跟分辨率有关）进行调整的。  
  
   那么绝对单位和相对单位是什么回事？首先可以了解下%单位。我们在学习导航栏的制作时，有过了解"一列宽度自适应"的布局手法。那里面将div容器的width属性设置为80%，用的就是相对单位---相对浏览器窗口的大小，div块的宽度是80%。  
  
   实际上我们可以这样理解，将px看做绝对单位---显示上的绝对单位。其他的相对单位都是以它为基础的。比如em,它就是相对当前文本字体的高度（假设当前文本字体的尺寸是 12px, 我们设置新的字体为1.5em,则新字体尺寸转换为 12 \* 1.5 = 18px）。要记住的就是em是相对父级元素的高度---假设第一个div里面我们设置了字体是12px,第二级我们设置为1.5em,第三级设置为1.5em，实际上字体的显示是：第一级12px,第二级18px，第三级 18 \* 1.5 = 27px。

# CSS3的文字阴影—text-shadow

前段时间整理了[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)中的**[渐变Gradient](http://www.w3cplus.com/node/44" \t "https://www.w3cplus.com/blog/_blank)**、**[透明度RGBA](http://www.w3cplus.com/node/45" \t "https://www.w3cplus.com/blog/_blank)**、**[边框圆角box-radius](http://www.w3cplus.com/node/48" \t "https://www.w3cplus.com/blog/_blank)**三个新属性的使用方法，这几次继续整理了有关于[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)的[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)的使用方法。今天特意花了点时间贴上来与有共同爱好的朋友一起分享这个新属性的应用。

[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)还没有出现时，大家在网页设计中阴影一般都是用photoshop做成图片，现在有了css3可以直接使用[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)属性来指定阴影。这个属性可以有两个作用，产生阴影和模糊主体。这样在不使用图片时能给文字增加质感。

text-shadow曾经在css2中就出现过，但在css2.1版本中又被抛弃了，现在[css3.0](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)版本又重新捡回来了。这说明text-shadow这个属性非常值得我们做前端的人员重视。现在很多项目中，[css3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)有很多属性被前端工程师们使用了，我在前面也分别介绍过[css3中的渐变](http://www.w3cplus.com/node/44" \t "https://www.w3cplus.com/blog/_blank)，[透明度](http://www.w3cplus.com/node/45" \t "https://www.w3cplus.com/blog/_blank)，[圆角](http://www.w3cplus.com/node/48" \t "https://www.w3cplus.com/blog/_blank)三个属性的使用，而且[css3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)这么多属性中，我个人觉得[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)运用得是最多的一个属性，作为我们前端人员我觉得有必要学习并掌握这个[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)属性。

****语法:****

 text-shadow ： none | <length> none | [<shadow>, ] \* <shadow> 或none | <color> [, <color> ]\*

也就是：

text-shadow:[颜色(Color) x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)],[颜色(color) x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)]...

或者

text-shadow:[x轴(X Offset) y轴(Y Offset) 模糊半径(Blur) 颜色(Color)],[x轴(X Offset) y轴(Y Offset) 模糊半径(Blur) 颜色(Color)]...

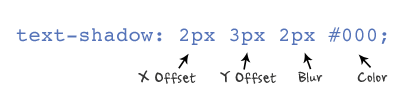
****取值:****

<length>：长度值，可以是负值。用来指定阴影的延伸距离。其中X Offset是水平偏移值，Y Offset是垂直偏移值

<color>：指定阴影颜色，也可以是rgba透明色

<shadow>：阴影的模糊值，不可以是负值，用来指定模糊效果的作用距离。

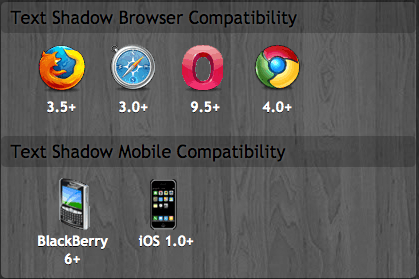
如下图所示：



****说明：****

可以给一个对象应用一组或多组阴影效果，方式如前面的语法显示一样，用逗号隔开。text-shadow: X-Offset Y-Offset Blur Color中X-Offset表示阴影的水平偏移距离，其值为正值时阴影向右偏移，如果其值为负值时，阴影向左偏移；Y-Offset是指阴影的垂直偏移距离，如果其值是正值时，阴影向下偏移反之其值是负值时阴影向顶部偏移；Blur是指阴影的模糊程度，其值不能是负值，如果值越大，阴影越模糊，反之阴影越清晰，如果不需要阴影模糊可以将Blur值设置为0；Color是指阴影的颜色，其可以使用rgba色。

****浏览器的兼容性：****



我们来看个实例

首先给所有的DEMO一个公用的样式和类名：

.demo {

background: #666666;

width: 440px;

padding: 30px;

font: bold 55px/100% "微软雅黑", "Lucida Grande", "Lucida Sans", Helvetica, Arial, Sans;;

color: #fff;

text-transform: uppercase;

}

接下来我们在每个Demo上加上自己特定的样式，分别如下所示：

 .demo1 {

text-shadow: red 0 1px 0;

}



效果是不错，可是让我们头痛的是IE是不支持text-shadow效果，但为了在兼容这一问题，我们只好使用滤镜filter:shadow来处理（本人不提倡使用滤镜）。filter:shadow滤镜作用与dropshadow类似，也能使用对象产生阴影效果，不同的是shadow可产生渐近效果，使用阴影更平滑实现。

****滤镜语法： E {filter:shadow(Color=颜色值,Direction=数值,Strength=数值)}****

其中E是元素选择器，Color用于设定对象的阴影色；Direction用于设定投影的主向，取值为0即零度（表示向上方向），45为右上，90为右，135为右下，180为下方，225为左下方，270为左方，315为左上方；Strength就是强度，类似于text-shadow中的blur值。

我们先不管IE下的效果了，我个人认为text-shadow运用好了，也能像photoshop一样制作出非常好的效果，下面我这里罗列出一些比较好看的实例以供大家参考

****注：下面所有的Demo需要加上这个公用的样式：****

****.demo {****

****background: #666666;****

****width: 440px;****

****padding: 30px;****

****font: bold 55px/100% "微软雅黑", "Lucida Grande", "Lucida Sans", Helvetica, Arial, Sans;;****

****color: #fff;****

****text-transform: uppercase;****

****}****

****效果一：Glow and Extra Glow effect(也就是NEON effect)****

 .demo2 {

text-shadow: 0 0 20px red;

}



辉光效果，我们设置比较大的模糊半径来增加其辉光效果，你可以改变不同的模糊半径值来达到不同的效果，当然你也可以同时增加几个不同的半径值，创造多种不同的阴影效果。就如下面的NEON效果。

 .demo3 {

text-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 15px #fff, 0 0 40px #ff00de, 0 0 70px #ff00de;

}



****效果二：Apple Style Effect****

.demo4 {

color: #000;

text-shadow: 0 1px 1px #fff;

}  


****效果三：Photoshop Emboss Effect****

 .demo5 {

color: #ccc;

text-shadow: -1px -1px 0 #fff,1px 1px 0 #333,1px 1px 0 #444;

}



效果二和效果三，我想用photoshop的朋友一定很熟悉，是不是很类似我们 photoshop中的投影和浮雕效果的呀。应用这两个效果大家一定要注意，其模糊值一定要设置为0，使文本不具有任何模糊效果，主要用来增加其质感，你也可以像photoshop中制作一样，改变不同的投光角度，从而制作出不同的效果，这里我就不举例子了，感兴趣的朋友可以自己尝试一下。

****效果四：Blurytext Effect****

 .demo6 {

color: transparent;

text-shadow: 0 0 5px #f96;

}



用text-shadow制作模糊的效果主要要注意一点就是，把文本的前景色设置为透明transparent,如果模糊值越大，其效果越糊糊；其二，我们不设置任何方向的偏移值。如果结合前面的photoshop emboss效果，可以让你等到不同的效果。提醒一下opera浏览器不支持这个效果。

我们结合前面的Photoshop Emboss效果，我们可以制作出一个带有模糊的浮雕效果：

 .demo7 {

color: transparent;

text-shadow:0 0 6px #F96, -1px -1px #FFF, 1px -1px #444;

}



****效果五：Inset text effect****1px 0 #E4F1FF;}

 .demo8 {

color: #566F89;

background: #C5DFF8;

text-shadow: 1px 1px 0 #E4F1FF;

}



这种效果需要注意以：文字的前景色要比背景色暗，阴影颜色稍比背景色亮一点点，这一步很重要，如果阴影色太亮看起来会怪，如果太暗将没有效果显示。具体实现可能看看[stylizedweb](http://stylizedweb.com/2009/10/22/how-to-create-inset-text-with-css3/" \t "https://www.w3cplus.com/blog/_blank)的制作。Inset效果是文本的影子效果，也是常见的一种效果，阴影是同一个小偏移量给人一种微妙的突出效果。

****效果六：Stroke text effect****px 0 #f96,-1px -1px 0 #f96; }

 .demo9 {

color: #fff;

text-shadow: 1px 1px 0 #f96,-1px -1px 0 #f96;

}



描边效果跟我们在Photoshop相比，我承认效果差很多，出现断点，但有时还是可以试用达到一种特殊的描边效果，其主要运用两个阴影，第一个向左上投影，而第二向右下投影，还需注意，制作描边的阴影效果我们不使用模糊值。

****效果七：3D text effect****

.demo10 {

color: #fff;

text-shadow: 1px 1px rgba(197, 223, 248,0.8),2px 2px rgba(197, 223, 248,0.8),3px 3px rgba(197, 223, 248,0.8),4px 4px rgba(197, 223, 248,0.8),5px 5px rgba(197, 223, 248,0.8),6px 6px rgba(197, 223, 248,0.8);

}



我们换过投影方向可以制作出另外一种3D文字效果

.demo11 {

color: #fff;

text-shadow: -1px -1px rgba(197, 223, 248,0.8),-2px -2px rgba(197, 223, 248,0.8),-3px -3px rgba(197, 223, 248,0.8),-4px -4px rgba(197, 223, 248,0.8),-5px -5px rgba(197, 223, 248,0.8),-6px -6px rgba(197, 223, 248,0.8);

}



3D文字效果运用原理就是像Photoshop一样，我们在文字的下方或上方复制了多个图层，并把每一个层向左上或右下方向移动一个1px距离，从而制作出3D效果。同时我们层数越多，其越厚重。换成用text-shadow制作就是使用多个阴影，并把阴影色设置相同，给其使用rgba色效果更佳，如上面的实例。

****效果八：Vintge/Retro text effect****

****.demo11 {****

****color: #eee;****

****text-shadow: 5px 5px 0 #666, 7px 7px 0 #eee;****

****}****



Vintage retro这种风格的文字效果是由两个文本阴影合成的，这里需要注意的是：第一个阴影色和背景色相同；文本前景色和第二个阴影色相同

****效果九：Anaglyphic text effect****

 .demo13 {

color: rgba(255, 179, 140,0.5);

text-shadow: 3px 3px 0 rgba(180,255,0,0.5);

}



anaglyphic文字效果起到一种补色的效果，从而制作出一种三维效果图。其效果是用css重新使用的文字阴影和文本前景的rgba色组合而成。在文本的前景色和阴影上同时使用rgba色，使底层的文字是通过影子可见。

上面我主要列出了九个不同效果的实例，当然大家可以改变各种方式制作出一些特殊的效果，也再一次证实[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)中的[text-shadow](http://www.w3.org/Style/Examples/007/text-shadow.en.html" \l "translations" \t "https://www.w3cplus.com/blog/_blank)实力是不浅的，希望大家能喜欢上这个属性，并能更好的使用这个属性。虽然现在IE不支持，但大家不要惧怕使用[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/blog/_blank)，因为我们做前端的迟早都会需要掌握这些新技术的。



浏览器兼容性



# CSS3 渐变（Gradients）

IMG_256

CSS3 渐变（gradients）可以让你在两个或多个指定的颜色之间显示平稳的过渡。

以前，你必须使用图像来实现这些效果。但是，通过使用 CSS3 渐变（gradients），你可以减少下载的事件和宽带的使用。此外，渐变效果的元素在放大时看起来效果更好，因为渐变（gradient）是由浏览器生成的。

CSS3 定义了两种类型的渐变（gradients）：

* **线性渐变（Linear Gradients）- 向下/向上/向左/向右/对角方向**
* **径向渐变（Radial Gradients）- 由它们的中心定义**

## 浏览器支持

表中的数字指定了完全支持该属性的第一个浏览器版本。

后边跟 -webkit-、-moz- 或 -o- 的数字指定了需加上前缀才能支持属性的第一个版本。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性 |  |  |  |  |  |
| linear-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| radial-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |
| repeating-linear-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| repeating-radial-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |

## CSS3 线性渐变

为了创建一个线性渐变，你必须至少定义两种颜色结点。颜色结点即你想要呈现平稳过渡的颜色。同时，你也可以设置一个起点和一个方向（或一个角度）。

**线性渐变的实例：**



### **语法**

background: linear-gradient(direction, color-stop1, color-stop2, ...);

**线性渐变 - 从上到下（默认情况下）**

下面的实例演示了从顶部开始的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从上到下的线性渐变：

#grad { background: -webkit-linear-gradient(red, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(red, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear" \t "http://www.runoob.com/css3/_blank)**

**线性渐变 - 从左到右**

下面的实例演示了从左边开始的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从左到右的线性渐变：

#grad { background: -webkit-linear-gradient(left, red , blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(right, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(right, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(to right, red , blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_ltr" \t "http://www.runoob.com/css3/_blank)**

**线性渐变 - 对角**

你可以通过指定水平和垂直的起始位置来制作一个对角渐变。

下面的实例演示了从左上角开始（到右下角）的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从左上角到右下角的线性渐变：

#grad { background: -webkit-linear-gradient(left top, red , blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(bottom right, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(bottom right, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(to bottom right, red , blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_diagonal" \t "http://www.runoob.com/css3/_blank)**

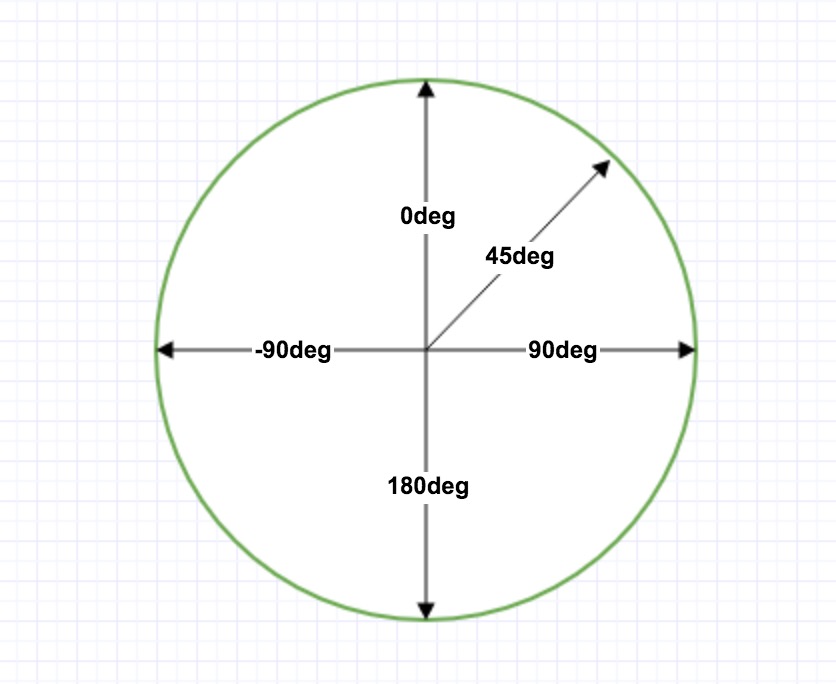
## 使用角度

如果你想要在渐变的方向上做更多的控制，你可以定义一个角度，而不用预定义方向（to bottom、to top、to right、to left、to bottom right，等等）。

### **语法**

background: linear-gradient(angle, color-stop1, color-stop2);

角度是指水平线和渐变线之间的角度，逆时针方向计算。换句话说，0deg 将创建一个从下到上的渐变，90deg 将创建一个从左到右的渐变。



但是，请注意很多浏览器(Chrome,Safari,fiefox等)的使用了旧的标准，即 0deg 将创建一个从左到右的渐变，90deg 将创建一个从下到上的渐变。换算公式 **90 - x = y** 其中 x 为标准角度，y为非标准角度。

下面的实例演示了如何在线性渐变上使用角度：

## 实例

带有指定的角度的线性渐变：

#grad { background: -webkit-linear-gradient(180deg, red, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(180deg, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(180deg, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(180deg, red, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_angles" \t "http://www.runoob.com/css3/_blank)**

## 使用多个颜色结点

下面的实例演示了如何设置多个颜色结点：

## 实例

带有多个颜色结点的从上到下的线性渐变：

#grad { background: -webkit-linear-gradient(red, green, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(red, green, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(red, green, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(red, green, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_cs" \t "http://www.runoob.com/css3/_blank)**

下面的实例演示了如何创建一个带有彩虹颜色和文本的线性渐变：

## 实例

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* Opera 11.1 - 12.0 \*/ background: -o-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* Firefox 3.6 - 15 \*/ background: -moz-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* 标准的语法 \*/ background: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet); }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_rainbow" \t "http://www.runoob.com/css3/_blank)**

## 使用透明度（transparent）

CSS3 渐变也支持透明度（transparent），可用于创建减弱变淡的效果。

为了添加透明度，我们使用 rgba() 函数来定义颜色结点。rgba() 函数中的最后一个参数可以是从 0 到 1 的值，它定义了颜色的透明度：0 表示完全透明，1 表示完全不透明。

下面的实例演示了从左边开始的线性渐变。起点是完全透明，慢慢过渡到完全不透明的红色：

## 实例

从左到右的线性渐变，带有透明度：

#grad { background: -webkit-linear-gradient(left,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Safari 5.1 - 6 \*/ background: -o-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Opera 11.1 - 12\*/ background: -moz-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Firefox 3.6 - 15\*/ background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_trans" \t "http://www.runoob.com/css3/_blank)**

## 重复的线性渐变

repeating-linear-gradient() 函数用于重复线性渐变：

## 实例

一个重复的线性渐变：

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Opera 11.1 - 12.0 \*/ background: -o-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Firefox 3.6 - 15 \*/ background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%); /\* 标准的语法 \*/ background: repeating-linear-gradient(red, yellow 10%, green 20%); }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_repeating" \t "http://www.runoob.com/css3/_blank)**

## CSS3 径向渐变

径向渐变由它的中心定义。

为了创建一个径向渐变，你也必须至少定义两种颜色结点。颜色结点即你想要呈现平稳过渡的颜色。同时，你也可以指定渐变的中心、形状（圆形或椭圆形）、大小。默认情况下，渐变的中心是 center（表示在中心点），渐变的形状是 ellipse（表示椭圆形），渐变的大小是 farthest-corner（表示到最远的角落）。

**径向渐变的实例：**



### **语法**

background: radial-gradient(center, shape size, start-color, ..., last-color);

**径向渐变 - 颜色结点均匀分布（默认情况下）**

## 实例

颜色结点均匀分布的径向渐变：

#grad { background: -webkit-radial-gradient(red, green, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(red, green, blue); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(red, green, blue); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(red, green, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial" \t "http://www.runoob.com/css3/_blank)**

**径向渐变 - 颜色结点不均匀分布**

## 实例

颜色结点不均匀分布的径向渐变：

#grad { background: -webkit-radial-gradient(red 5%, green 15%, blue 60%); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(red 5%, green 15%, blue 60%); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(red 5%, green 15%, blue 60%); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(red 5%, green 15%, blue 60%); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial2" \t "http://www.runoob.com/css3/_blank)**

## 设置形状

shape 参数定义了形状。它可以是值 circle 或 ellipse。其中，circle 表示圆形，ellipse 表示椭圆形。默认值是 ellipse。

## 实例

形状为圆形的径向渐变：

#grad { background: -webkit-radial-gradient(circle, red, yellow, green); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(circle, red, yellow, green); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(circle, red, yellow, green); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(circle, red, yellow, green); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial_shape" \t "http://www.runoob.com/css3/_blank)**

## 不同尺寸大小关键字的使用

size 参数定义了渐变的大小。它可以是以下四个值：

* **closest-side**
* **farthest-side**
* **closest-corner**
* **farthest-corner**

## 实例

带有不同尺寸大小关键字的径向渐变：

#grad1 { /\* Safari 5.1 - 6.0 \*/ background: -webkit-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* Opera 11.6 - 12.0 \*/ background: -o-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* Firefox 3.6 - 15 \*/ background: -moz-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* 标准的语法 \*/ background: radial-gradient(60% 55%, closest-side,blue,green,yellow,black); } #grad2 { /\* Safari 5.1 - 6.0 \*/ background: -webkit-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* Opera 11.6 - 12.0 \*/ background: -o-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* Firefox 3.6 - 15 \*/ background: -moz-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* 标准的语法 \*/ background: radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); }

## 重复的径向渐变

repeating-radial-gradient() 函数用于重复径向渐变：

## 实例

一个重复的径向渐变：

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Opera 11.6 - 12.0 \*/ background: -o-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Firefox 3.6 - 15 \*/ background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%); /\* 标准的语法 \*/ background: repeating-radial-gradient(red, yellow 10%, green 15%);

# CSS3 渐变（Gradients）

IMG_256

CSS3 渐变（gradients）可以让你在两个或多个指定的颜色之间显示平稳的过渡。

以前，你必须使用图像来实现这些效果。但是，通过使用 CSS3 渐变（gradients），你可以减少下载的事件和宽带的使用。此外，渐变效果的元素在放大时看起来效果更好，因为渐变（gradient）是由浏览器生成的。

CSS3 定义了两种类型的渐变（gradients）：

* **线性渐变（Linear Gradients）- 向下/向上/向左/向右/对角方向**
* **径向渐变（Radial Gradients）- 由它们的中心定义**

## 浏览器支持

表中的数字指定了完全支持该属性的第一个浏览器版本。

后边跟 -webkit-、-moz- 或 -o- 的数字指定了需加上前缀才能支持属性的第一个版本。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性 |  |  |  |  |  |
| linear-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| radial-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |
| repeating-linear-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| repeating-radial-gradient | 10.0 | 26.0 10.0 -webkit- | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |

## CSS3 线性渐变

为了创建一个线性渐变，你必须至少定义两种颜色结点。颜色结点即你想要呈现平稳过渡的颜色。同时，你也可以设置一个起点和一个方向（或一个角度）。

**线性渐变的实例：**



### **语法**

background: linear-gradient(direction, color-stop1, color-stop2, ...);

**线性渐变 - 从上到下（默认情况下）**

下面的实例演示了从顶部开始的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从上到下的线性渐变：

#grad { background: -webkit-linear-gradient(red, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(red, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear" \t "http://www.runoob.com/css3/_blank)**

**线性渐变 - 从左到右**

下面的实例演示了从左边开始的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从左到右的线性渐变：

#grad { background: -webkit-linear-gradient(left, red , blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(right, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(right, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(to right, red , blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_ltr" \t "http://www.runoob.com/css3/_blank)**

**线性渐变 - 对角**

你可以通过指定水平和垂直的起始位置来制作一个对角渐变。

下面的实例演示了从左上角开始（到右下角）的线性渐变。起点是红色，慢慢过渡到蓝色：

## 实例

从左上角到右下角的线性渐变：

#grad { background: -webkit-linear-gradient(left top, red , blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(bottom right, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(bottom right, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(to bottom right, red , blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_diagonal" \t "http://www.runoob.com/css3/_blank)**

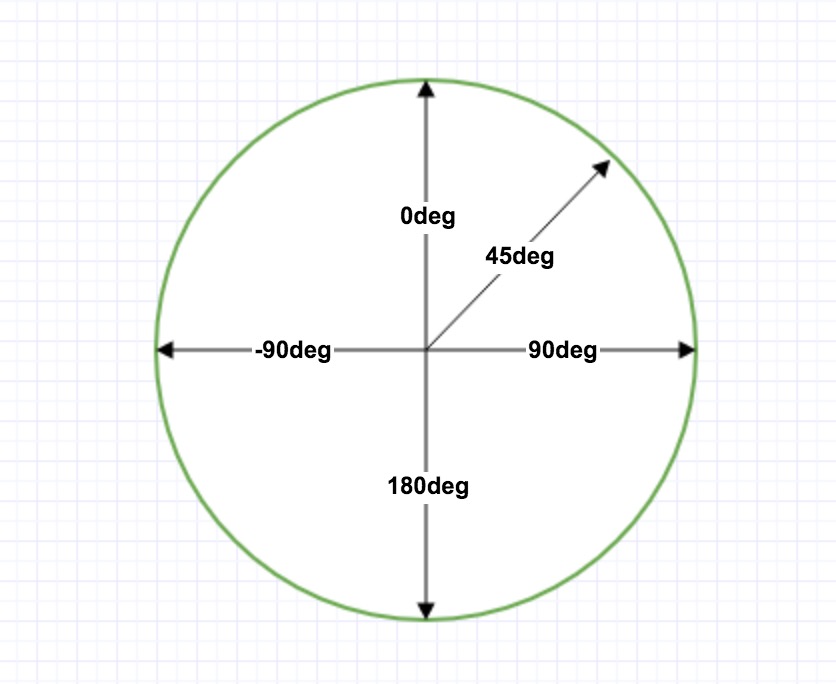
## 使用角度

如果你想要在渐变的方向上做更多的控制，你可以定义一个角度，而不用预定义方向（to bottom、to top、to right、to left、to bottom right，等等）。

### **语法**

background: linear-gradient(angle, color-stop1, color-stop2);

角度是指水平线和渐变线之间的角度，逆时针方向计算。换句话说，0deg 将创建一个从下到上的渐变，90deg 将创建一个从左到右的渐变。



但是，请注意很多浏览器(Chrome,Safari,fiefox等)的使用了旧的标准，即 0deg 将创建一个从左到右的渐变，90deg 将创建一个从下到上的渐变。换算公式 **90 - x = y** 其中 x 为标准角度，y为非标准角度。

下面的实例演示了如何在线性渐变上使用角度：

## 实例

带有指定的角度的线性渐变：

#grad { background: -webkit-linear-gradient(180deg, red, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(180deg, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(180deg, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(180deg, red, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_angles" \t "http://www.runoob.com/css3/_blank)**

## 使用多个颜色结点

下面的实例演示了如何设置多个颜色结点：

## 实例

带有多个颜色结点的从上到下的线性渐变：

#grad { background: -webkit-linear-gradient(red, green, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(red, green, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(red, green, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(red, green, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_cs" \t "http://www.runoob.com/css3/_blank)**

下面的实例演示了如何创建一个带有彩虹颜色和文本的线性渐变：

## 实例

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* Opera 11.1 - 12.0 \*/ background: -o-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* Firefox 3.6 - 15 \*/ background: -moz-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet); /\* 标准的语法 \*/ background: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet); }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_rainbow" \t "http://www.runoob.com/css3/_blank)**

## 使用透明度（transparent）

CSS3 渐变也支持透明度（transparent），可用于创建减弱变淡的效果。

为了添加透明度，我们使用 rgba() 函数来定义颜色结点。rgba() 函数中的最后一个参数可以是从 0 到 1 的值，它定义了颜色的透明度：0 表示完全透明，1 表示完全不透明。

下面的实例演示了从左边开始的线性渐变。起点是完全透明，慢慢过渡到完全不透明的红色：

## 实例

从左到右的线性渐变，带有透明度：

#grad { background: -webkit-linear-gradient(left,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Safari 5.1 - 6 \*/ background: -o-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Opera 11.1 - 12\*/ background: -moz-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\* Firefox 3.6 - 15\*/ background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_trans" \t "http://www.runoob.com/css3/_blank)**

## 重复的线性渐变

repeating-linear-gradient() 函数用于重复线性渐变：

## 实例

一个重复的线性渐变：

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Opera 11.1 - 12.0 \*/ background: -o-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Firefox 3.6 - 15 \*/ background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%); /\* 标准的语法 \*/ background: repeating-linear-gradient(red, yellow 10%, green 20%); }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-linear_repeating" \t "http://www.runoob.com/css3/_blank)**

## CSS3 径向渐变

径向渐变由它的中心定义。

为了创建一个径向渐变，你也必须至少定义两种颜色结点。颜色结点即你想要呈现平稳过渡的颜色。同时，你也可以指定渐变的中心、形状（圆形或椭圆形）、大小。默认情况下，渐变的中心是 center（表示在中心点），渐变的形状是 ellipse（表示椭圆形），渐变的大小是 farthest-corner（表示到最远的角落）。

**径向渐变的实例：**



### **语法**

background: radial-gradient(center, shape size, start-color, ..., last-color);

**径向渐变 - 颜色结点均匀分布（默认情况下）**

## 实例

颜色结点均匀分布的径向渐变：

#grad { background: -webkit-radial-gradient(red, green, blue); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(red, green, blue); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(red, green, blue); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(red, green, blue); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial" \t "http://www.runoob.com/css3/_blank)**

**径向渐变 - 颜色结点不均匀分布**

## 实例

颜色结点不均匀分布的径向渐变：

#grad { background: -webkit-radial-gradient(red 5%, green 15%, blue 60%); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(red 5%, green 15%, blue 60%); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(red 5%, green 15%, blue 60%); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(red 5%, green 15%, blue 60%); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial2" \t "http://www.runoob.com/css3/_blank)**

## 设置形状

shape 参数定义了形状。它可以是值 circle 或 ellipse。其中，circle 表示圆形，ellipse 表示椭圆形。默认值是 ellipse。

## 实例

形状为圆形的径向渐变：

#grad { background: -webkit-radial-gradient(circle, red, yellow, green); /\* Safari 5.1 - 6.0 \*/ background: -o-radial-gradient(circle, red, yellow, green); /\* Opera 11.6 - 12.0 \*/ background: -moz-radial-gradient(circle, red, yellow, green); /\* Firefox 3.6 - 15 \*/ background: radial-gradient(circle, red, yellow, green); /\* 标准的语法 \*/ }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial_shape" \t "http://www.runoob.com/css3/_blank)**

## 不同尺寸大小关键字的使用

size 参数定义了渐变的大小。它可以是以下四个值：

* **closest-side**
* **farthest-side**
* **closest-corner**
* **farthest-corner**

## 实例

带有不同尺寸大小关键字的径向渐变：

#grad1 { /\* Safari 5.1 - 6.0 \*/ background: -webkit-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* Opera 11.6 - 12.0 \*/ background: -o-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* Firefox 3.6 - 15 \*/ background: -moz-radial-gradient(60% 55%, closest-side,blue,green,yellow,black); /\* 标准的语法 \*/ background: radial-gradient(60% 55%, closest-side,blue,green,yellow,black); } #grad2 { /\* Safari 5.1 - 6.0 \*/ background: -webkit-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* Opera 11.6 - 12.0 \*/ background: -o-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* Firefox 3.6 - 15 \*/ background: -moz-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); /\* 标准的语法 \*/ background: radial-gradient(60% 55%, farthest-side,blue,green,yellow,black); }

**[尝试一下 »](http://www.runoob.com/try/try.php?filename=trycss3_gradient-radial_size" \t "http://www.runoob.com/css3/_blank)**

## 重复的径向渐变

repeating-radial-gradient() 函数用于重复径向渐变：

## 实例

一个重复的径向渐变：

#grad { /\* Safari 5.1 - 6.0 \*/ background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Opera 11.6 - 12.0 \*/ background: -o-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Firefox 3.6 - 15 \*/ background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%); /\* 标准的语法 \*/ background: repeating-radial-gradient(red, yellow 10%, green 15%); }



# CSS3 box-shadow

作者：大漠 日期：2011-05-11 点击：118969

[box-shadow](https://www.w3cplus.com/blog/tags/28.html" \t "https://www.w3cplus.com/content/_blank)

[css3属性详解](https://www.w3cplus.com/blog/tags/250.html" \t "https://www.w3cplus.com/content/_blank)

编辑推荐： [掘金](https://juejin.im/?utm_source=w3cplus&utm_medium=word&utm_campaign=q1_website" \t "https://www.w3cplus.com/content/_blank)是一个高质量的技术社区，从 CSS 到 Vue.js，性能优化到开源类库，让你不错过前端开发的每一个技术干货。 [点击链接](https://juejin.im/?utm_source=w3cplus&utm_medium=word&utm_campaign=q1_website" \t "https://www.w3cplus.com/content/_blank)查看最新前端内容，或到各大应用市场搜索「 [掘金](https://juejin.im/?utm_source=w3cplus&utm_medium=word&utm_campaign=q1_website" \t "https://www.w3cplus.com/content/_blank)」下载APP，技术干货尽在掌握中。

上一节我们一起探讨了一下[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的文字阴影[text-shadow](http://www.w3cplus.com/node/52" \t "https://www.w3cplus.com/content/_blank)的使用方法，今天我们接着一起来探讨一下[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)中的另一个属性[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)的使用方法。[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)有点类似于[text-shadow](http://www.w3cplus.com/node/52" \t "https://www.w3cplus.com/content/_blank)，只不过不同的是[text-shadow](http://www.w3cplus.com/node/52" \t "https://www.w3cplus.com/content/_blank)是对象的文本设置阴影，而[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)是给对象实现图层阴影效果。

本文我们搁下IE不谈，只谈谈[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)的具体使用方法

****语法：****

E {box-shadow: <length> <length> <length>?<length>?||<color>}

也就是：

E {box-shadow:inset x-offset y-offset blur-radius spread-radius color}

换句说：

对象选择器 {box-shadow:投影方式 X轴偏移量 Y轴偏移量 阴影模糊半径 阴影扩展半径 阴影颜色}

box-shadow和text-shadow一样可以使用一个或多个投影，如果使用多个投影时必须需要用逗号“，”分开。

****取值：****

box-shadow属性至多有6个参数设置，他们分别取值：

****阴影类型：****此参数是一个可选值，如果不设值，其默认的投影方式是外阴影；如果取其唯一值“inset”,就是将外阴影变成内阴影，也就是说设置阴影类型为“inset”时，其投影就是内阴影；

****X-offset:****是指阴影水平偏移量其值可以是正负值可以取正负值，如果值为正值，则阴影在对象的右边，反之其值为负值时，阴影在对象的左边；

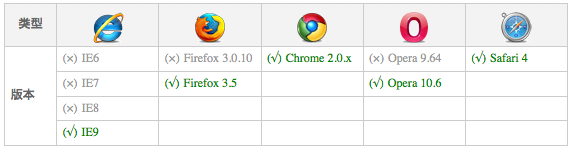
****Y-offset:****是指阴影的垂直偏移量，其值也可以是正负值，如果为正值，阴影在对象的底部，反之其值为负值时，阴影在对象的顶部；

****阴影模糊半径：****此参数是可选，，但其值只能是为正值，如果其值为0时，表示阴影不具有模糊效果，其值越大阴影的边缘就越模糊；

****阴影扩展半径：****此参数可选，其值可以是正负值，如果值为正，则整个阴影都延展扩大，反之值为负值是，则缩小

****阴影颜色：****此参数可选，如果不设定任何颜色时，浏览器会取默认色，但各浏览器默认色不一样，特别是在webkit内核下的safari和chrome浏览器将无色，也就是透明，建议不要省略此参数。

****浏览器的兼容：****



我们这里还涉及到一个各浏览器前缀的问题，比如说Mozilla内核的-moz，webkit内核的-webkit。经测试在最新版的Firefox和Google Chrome浏览器都无需加上前缀，但在safari中还是需要前缘的，为了能兼容支持的各大浏览器，我们在书写box-shadow的格式应该这样

 //Firefox4.0-

-moz-box-shadow: 投影方式 X轴偏移量 Y轴偏移量 阴影模糊半径 阴影扩展半径 阴影颜色;

//Safari and Google chrome10.0-

-webkit-box-shadow: 投影方式 X轴偏移量 Y轴偏移量 阴影模糊半径 阴影扩展半径 阴影颜色;

//Firefox4.0+ 、 Google chrome 10.0+ 、 Oprea10.5+ and IE9

box-shadow: 投影方式 X轴偏移量 Y轴偏移量 阴影模糊半径 阴影扩展半径 阴影颜色;

****box-shadow的特征：****

较之ps制作出来的图片相比，CSS3的box-shadow可以通过改变其参数得到不同的效果，如：改变阴影偏移量的设置，我们可以使用阴影只在对象的上下左右的任一边出现，也可以让其出现在其中的某几个边上;其二可以随时调节阴影大小，边缘模糊度，阴影颜色，其三可以随时更改为内阴影，另外还可以设置多个阴影效果

我们先来看一个简单的实例：

.demo1 {

-webkit-box-shadow: 3px 3px 3px;

-moz-box-shadow: 3px 3px 3px;

box-shadow: 3px 3px 3px;

}

Firefox/Opera下效果   IMG_258Safari/Chrome下效果

回到上面那个实例，其实在webkit内核的浏览器Safari、Google Chrome里不会有任何阴影效果，虽然W3C标准里说颜色是可选择的，但是在没有给出颜色的时候 ，safari/chrome和firefox表现不同，在webkit内核的浏览器下阴影表现为透明色而mozilla和oprea下表现为黑色。基于这样的原因，大家在使用box-shadow时不要忘了加上阴影颜色的值。

根据上面的现像，我们来看一个box-shadow有关阴影是否会被计算为内容的实例。

 <div class="outer">

<div class="inter"> </div>

</div>

我们把外面div设置为100px\*100px,里面div设置为60px\*60px，并在里面的div上加上一个向下向右偏移50px的绿色阴影，我们看看多出来的阴影会怎么样？

.outer {

width: 100px;

height: 100px;

border: 1px solid #ccc;

}

.inter {

width: 60px;

height: 60px;

margin: 10px auto;

background: #f69;

-webkit-box-shadow: 50px 50px green;

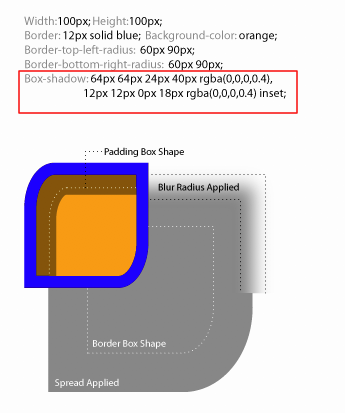
-moz-box-shadow: 50px 50px green;

box-shadow: 50px 50px green;

}



从各大浏览中的效果我们可以看出，阴影多出来的阴影会撑破容器跑出来。标准里有一张图，描述了box-shadow的工作方式，这张图直观告诉我们如何使用box-shadow



这张图可以告诉我们很多信息，比如说borer-radius圆角，阴影扩展、阴影模糊以及padding是如何影响对象阴影的：非零值的border-radius将会以相同的作用影响阴影的外形，但border-image不会影响对象阴影的任何外形；对象阴影同box模型的层次一样，外阴影会在对象背景之下，内阴影会在边框之下背景之上。所以整个层级就是：边框>内阴影>背景图片>背景颜色>外阴影。因为大家都知道，我们的背景图片是在背景颜色之上的。

****IE滤镜方法：****

在前面我们讲过，IE9以下是不支持CSS3的box-shadow的，但为了处理这个兼容问题，我们可以在IE下使用IE的shadow阴影滤镜来实现

filter: progid:DXImageTransform.Microsoft.Shadow(color=’颜色值’, Direction=阴影角度（数值）, Strength=阴影半径（数值）);

注意：该[滤镜](http://msdn.microsoft.com/en-us/library/ms532979(VS.85).aspx" \t "https://www.w3cplus.com/content/_blank)必须配合background属性一起使用，否则该滤镜失效。除了使用[滤镜](http://msdn.microsoft.com/en-us/library/ms532979(VS.85).aspx" \t "https://www.w3cplus.com/content/_blank)的方法外，我们还有一种方法可以实现IE下的效果。那就是使用jQuery的插件[jquery.boxshadow.js](http://www.hintzmann.dk/testcenter/js/jquery/boxshadow/jquery.boxshadow.js" \t "https://www.w3cplus.com/content/_blank)。那么具体如何使用呢？其实很简单，你先下载这个[jquery.boxshadow.js](http://www.opensource.org/licenses/mit-license.php" \t "https://www.w3cplus.com/content/_blank)插件到你的项目中，接着把jquery版本库和jquery.boxshadow.js加载到页面上，如

<script type="text/javascript" src="../js/jquery.min.js"></script>

<script type="text/javascript" src="../js/jquery.boxshadow.js"></script>

然后你可以创建一个单独的js文件来处理，或者直接在页面的<head></head>里欠入一个<script> </script>，我们这里就只例出一个直接在head插入的解决办法：

 $(document).ready(function(){

if($.browser.msie) {

$('.demo1').boxShadow(0,0,5,"#888"); //demo1元素使用了box-shadow

$('.demo2').boxShadow(-10,-10,5,"#f36"); //demo2元素使用了box-shadow

}

});

上面我们了解了CSS3的box-shadow相关基础知识，那么下面我们通过一些实例来巩固一下box-shadow的具体用法：

如果没有进行特殊说明，我们这里的实例所用的HTML代码都如下，只是改变第二个class名称，如demo1 demo2等：

<div class="demo demo2></div>

基本的CSS样式

.demo {

width: 100px;

height: 50px;

background: #f69;

}

提醒大家：为了节约时间，下面的css代码中我只写了一个box-shadow，但是大家在实际应用中一定要记得把:-webkit-box-shadow和-moz-box-shadow加上去，不然在safari和chrome浏览器下是会没有任何效果的，这个我们在前面提过，此处不详说。

****效果一:单边效果****

  .dome2 {

box-shadow: -2px 0 0 green, //左边阴影

0 -2px 0 blue, //顶部阴影

0 2px 0 red, //底部阴影

2px 0 0 yellow; //右边阴影

}



上例中，我们分别对对象的四个边进行了box-shadow的设置，只不过我们使用了多层次的box-shadow应用，如果只需要在对象某一边应用阴影时，我们可以删除不使用阴影的设置。给对象四边设计阴影，我们是通过改变x-offset和y-offset的正负值来实现，其中x-offset为负值时，生成左边阴影，为正值时生成右边阴影，y-offset为正值是生成底部阴影，为负值时生成顶部阴影。并且把模糊半径设置为0，如果不设置为0的话那么其他三边也将会有阴影，并且此处还涉及到一个多阴影的顺序问题。当给同一个元素使用多个阴影属性时，需要注意它的顺序，最先写的阴影将显示在最顶层，如我们将上面的实例变一下，给其加上模糊值，将更能看出效果：

  .demo3 {

box-shadow: -2px 0 5px green,0 -2px 5px blue,0 2px 5px red,2px 0 5px yellow;

}



这样我们上例中：左边的放在了第一，其green阴影色在顶边的blue上，而顶边的blue在又在右边的yellow上，右这的yellow却在底边的red上。所以应用多次阴影的写法一定要注意其顺序问题，特别的当阴影的模糊值不一样的情况之下，另外有些网站介绍说可以写成下面的形式，但我经过多个浏览器测试，这种写法是无效的，

  .demo4 {

/\*这种写法是错误的（，网上有介绍说可以这样书写，但我测试多次未见效果，所以本人提倡不要这样书次，以免造成不必要的错误）\*/

box-shadow: -2px 0 0 green,box-shadow: 0 -2px 0 blue,box-shadow: 0 2px 0 red,box-shadow: 2px 0 0 yellow;

}

IMG_263

从上图的效果中也再一次证明了上面的写法是不正确的，希望大家在实际应用中时一定要注意多层次阴影的书写方法。同时也提醒大家在网上看相关资料时一定不能尽信，最好是能自己抽空验正一下。

在使用多层次的阴影时还需注意一个细节问题，如果前面的阴影模糊值小于后面的阴影模糊值，那么前面的显示在后面之上，如果前面阴影的模糊值大于后面的阴影模糊值，那么前面的阴影将遮住后面的阴影效果。如下面例子：

  /\*第一个阴影模糊半径值小于第二阴影模糊半径\*/

.demo5 {

box-shadow: 0 0 5px red,0 0 15px blue;

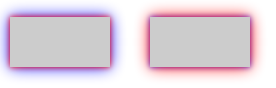
}

/第一个阴影模糊半径大于第二阴影模糊半径\*/

.demo6 {

box-shadow: 0 0 15px red, 0 0 5px blue;

}



实例效果再次证明：左图中我们可以看见红色阴影在兰色阴影之上并没有遮盖蓝色阴影，因为我们红色的阴影模糊值只有5px，比蓝色的15px模糊值要小；而右图中我们只能看到红色的阴影，那是因为我们第一个红色阴影的模糊半径大于第二个兰色的模糊半径，所以红色的阴影把蓝色的阴影遮盖住了。这一点大家可记住了。

****效果二：四边具有相同的阴影效果（只设置阴影模糊半径和阴影颜色）****

.demo7 {

box-shadow: 0 0 5px rgb(250,0,0);

}



我们在这里设置的是HEX值，我们也可以应用css3的rgba值给box-shadow的阴影颜色上，这样的好处是，box-shadow阴影色多了一个alpha透明值 ，如下面的实例：

.demo8 {

box-shadow: 0 0 5px rgba(250,0,0,0.5);

}



对比上面两个例子，前一个例子我们没有应用透明值，而后面一个例子我们应用了0.5的透明值，相比之下后面的阴影是不是更浅。当然在实践应用中您可以根据自己的需求进行设置。

****效果三：四边具有相同的阴影（只设置阴影扩展半径和阴影颜色）****

 .demo9 {

box-shadow: 0 0 0 1px red;

}



从效果中大家想想这种效果是不是跟我们在元素中的boder: 1px solid red;属性产生的效果很相似的呀。对的，box-shadow不单可以制作出阴影的效果，我们还可以利用其扩展半径这个值，来给对象制作出类似于边框的样式。下面我们来看一个对比的实例： 1}

  /\*边框效果\*/

.demo10 {

border: 1px solid red;

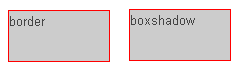
}

/\*阴影效果\*/

.demo11 {

box-shadow: 0 0 0 1px red;

}



实际上利用box-shadow来制作边框，只能说看上去像边框，但实质其并非边框，他和border还是有本质上的区别。从上面的效果图中我们明显的可以看出左边的box要比右边的box低那么1px的，这样一来随着其扩展半径值越大，两者之间的相差就更大，如：

 .demo12 {

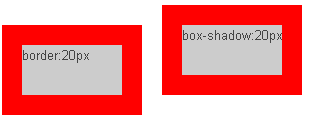
border: 20px solid red;

}

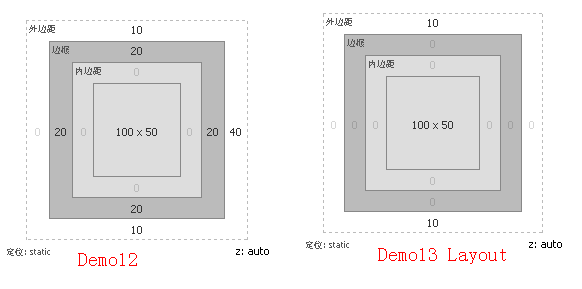
.demo13 {

box-shadow: 0 0 0 20px red;

}



我们接着来看demo12和demo13两个demo在firebug下的layout图：



结合上图两者在firebug下的layout图，更证实了我们前面所讲的阴影不会影响页面的任何布局：demo12的边框被计算了宽度，但demo13的阴影浏览器却忽略不计，所以借住这个特点，我们阴影所模拟的边框理可以自由的使用，但必须要注意其层级关系。

前面我们主要举例说明了如何利用box-shadow给对象****单边加上阴影效果****、****多边应用阴影效果****、****四边同时应用相同的阴影效果****以及如何应用****阴影模仿对象边框效果****等，这些都是我们box-shadow常用的一些阴影效果，下面我们在来例举几个特殊的实例：****内阴影inset****、****body设置阴影****和****投影drop shadow****。

****内阴影inset效果：****

 .demo14 {

box-shadow: inset 0 0 10px red;

}



上图中我们实现了div上添加内阴影的效果，我们这里要提醒一点的是，img标签上直接应用box-shadow的inset是没有任何效果的，为了证实这一点，我们一起来看下面的一个实例：

  <img src="/images/box-shadow-img.png" alt="box shadow img" />

.img-wrap {

-webkit-box-shadow: inset 0 0 10px red;

-moz-box-shadow: inset 0 0 10px red;

box-shadow: inset 0 0 10px red;

display: inline-block;

}

.img-wrap img {

position: relative;

z-index: -1;

}



上面的效果图再次证实了我们前面所说的，直接在img元素上使用inset是没有任何效果的，那么我们现在来针对这个bug做一次修改，我们把img放到一个div中，然后不直接在img上运用box-shadow属性，而是在img的父元素div上运用box-shadow，接着我们在给img进行相对定位，并让其在父元素下一层，如：

  <div class="shadow"><img src="/images/box-shadow-img.png" alt="box shadow img" /></div>

我们来看其主要的样式：

  .shadow {

position: relative;

display: inline-block;

\*display: inline;

}

.shadow::before {

content:"";

position: absolute;

width: 100%;

height: 100%;

-moz-box-shadow:inset 0 0 5px 1px red;

-webkit-box-shadow: inset 0 0 5px 1px red;

box-shadow: inset 0 0 5px 1px red;

}



根据上面实例思路，我们换过一种实现方法，这种方法我们是在img父元素上应用一上伪元素“:before”来实现：

 <div class="shadow"><img src="/images/box-shadow-img.png" alt="box shadow img" /></div>

.shadow {

position: relative;

display: inline-block;

\*display: inline;

}

.shadow::before {

content:"";

position: absolute;

width: 100%;

height: 100%;

-moz-box-shadow:inset 0 0 5px 1px red;

-webkit-box-shadow: inset 0 0 5px 1px red;

box-shadow: inset 0 0 5px 1px red;}



从效果上看，我们是不是同样实现了img加box-shadow的inset阴影呀，最后我们在来利用jQuery来解决img内阴影，这种方法的原理是我们通过jQuery把img标签转换成一个div元素，同时把img转换成div元素的背景图片，然后在这个div元素上应用内阴影，因为我们都知道div上应用内阴影是没有任何问题的。下面我们就一起来看其实现的方法

Html Code:

<img src="/images/box-shadow-img.png" alt="" class="inset-shadow" />

Css Code:

.inset-shadow{

-moz-box-shadow: 0 0 5px red inset;

-webkit-box-shadow: 0 0 5px red inset;

box-shadow: 0 0 5px red inset;}

jQuery code

  <script type="text/javascript">

$(document).ready(function(){

$('img.inset-shadow').each(function(){

var $img = $(this);

$img.load(function(){

var $div = $('<div/>');

$div.width($img.width());

$div.height($img.height());

$div.css('background-image', 'url('+$img.attr('src')+')');

var display = $img.css('display');

//If the div is set to inline the width and height will be 0 :(

//inline-block appears to be the only way around it but it's not

//supported in all browsers :( The browsers it's not supported in

//are probably the same ones that don't support box-shadow,

//so a solution maybe to add a browser check.

if(display === 'inline'){

$div.css('display', 'inline-block');

}else{

$div.css('display', display);

}

$div.attr('class', $img.attr('class'));

$img.replaceWith($div);

});

});

});

</script>



大家可以通过自己的firebug查看其中img的变化。

IMG_276

有关img上使用内阴影的使用方法，你可以参阅[CSS Box-Shadow:Inset](http://trentwalton.com/2010/11/22/css-box-shadowinset/" \t "https://www.w3cplus.com/content/_blank)一文。

****给body顶部增加一个阴影****

body:before {

content:"";

position:fixed;

top: -10px;

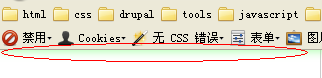
left: 0;

width: 100%;

height: 10px;

z-index: 999;

box-shadow: 0 0 10px rgba(125,255,125,0.8);}



这里提醒大家，为了不影响布局，top的取值最好和height的取值一致。只是top使用负值。利用同样的方法我们可以给任何一个元素加上阴影，但相应需要改变定位方式兴。

****Drop-shadow效果****

Drop-shadow效果，大家在Photoshop中肯定都见识过了，今天我们是来看一个实例，不增加任何元素标签的情况下，我们主要是利用box-shadow配合元素的两个伪元素:before和:after以及定位来实现，这种效果支持的浏览器现在主要有firefox3.5+/chrome5+/safari5+/opera10.6+/Ie9+。

我们先来了解一下其原理：我们通过box-shadow实现drop shadow效果是仅用一个div标签元素，然后配合其两个伪元素":before"和":after"；然后我们分别给其伪元素定位到div的后面，并把box-shadow应用到这两个伪元素上。具体我们来看其实现步骤：

先来看其html代码：

<div class="drop-shadow">drop shadow effect</div>

我定义了一个叫"drop-shadow"的div，现在我们给其应用一个基本样式

.drop-shadow {

width: 300px;

height: 150px;

position: relative;

background: #ccc;

}

接着我们给drop-shadow的“:before”和":after"定位到drop-shadow下面：

.drop-shadow:before,

.drop-shadow:after {

content: "";

position: absolute;

z-index: -1;

bottom: 15px;

left: 10px;

width: 50%;

height: 20%;

}

给drop-shadow的":before"和":after"加上阴影效果

.drop-shadow:before,

.drop-shadow:after {

content: "";

position: absolute;

z-index: -1;

bottom: 15px;

left: 10px;

width: 50%;

height: 20%;

/\*add box-shadow\*/

-webkit-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

-moz-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

box-shadow: 0 15px 10px rgba(125,125,125,0.8);

}

现在我们只得到了一边的阴影效果，那我么们可以通过应用[css3 transforms](http://www.w3.org/TR/css3-2d-transforms/" \t "https://www.w3cplus.com/content/_blank)来实现另一边的效果（有关CSS3的[transform](http://www.w3.org/TR/css3-2d-transforms/" \t "https://www.w3cplus.com/content/_blank)属性使用，我们将在下文介绍）

 .drop-shadow:before,

.drop-shadow:after {

content: "";

position: absolute;

z-index: -1;

bottom: 15px;

left: 10px;

width: 50%;

height: 20%;

/\*add box-shadow\*/

-webkit-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

-moz-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

box-shadow: 0 15px 10px rgba(125,125,125,0.8);

/\*add css3 transform\*/

-webkit-transform: rotate(-3deg);

-moz-transform: rotate(-3deg);

-o-transform: rotate(-3deg);

transform: rotate(-3deg);

}

我们现在只需要改变":after"伪元素定位方向。（伪元素":after"在相反方向旋转，相对于":before"）

.drop-shadow:after {

right:10px;

left: auto;

-webkit-transform:rotate(3deg);

-moz-transform:rotate(3deg);

-o-transform:rotate(3deg);

transform:rotate(3deg);

}

Drop shadow最终核心代码如下所示，只是我们在“:before”和“:after”中加了一个"max-width":的限制，

.drop-shadow {

width: 300px;

height: 150px;

position: relative;

background: #ccc;

margin-left: 100px;

}

.drop-shadow:before,

.drop-shadow:after {

content: "";

position: absolute;

z-index: -1;

bottom: 15px;

left: 10px;

width: 50%;

max-width: 150px;

height: 20%;

/\*add box-shadow\*/

-webkit-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

-moz-box-shadow: 0 15px 10px rgba(125,125,125,0.8);

box-shadow: 0 15px 10px rgba(125,125,125,0.8);

/\*add css3 transform\*/

-webkit-transform: rotate(-3deg);

-moz-transform: rotate(-3deg);

-o-transform: rotate(-3deg);

transform: rotate(-3deg);

}

.drop-shadow:after {

right:10px;

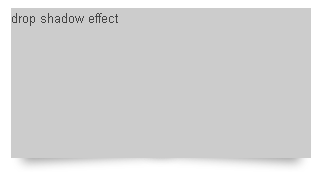
left: auto;

-webkit-transform:rotate(3deg);

-moz-transform:rotate(3deg);

-o-transform:rotate(3deg);

transform:rotate(3deg);

}  


有关drop shadow的更多的[demo](http://nicolasgallagher.com/css-drop-shadows-without-images/demo/" \t "https://www.w3cplus.com/content/_blank)和[Pure CSS3 box-shadow page curl effect](http://matthamm.com/box-shadow-curl.html" \t "https://www.w3cplus.com/content/_blank)可以查阅。如果对drop shadow感兴趣，大家可以点击：[Drop Shadows with CSS3](http://nimbupani.com/drop-shadows-with-css3.html" \t "https://www.w3cplus.com/content/_blank)和[CSS drop-shadows without images](http://nicolasgallagher.com/css-drop-shadows-without-images/" \t "https://www.w3cplus.com/content/_blank)查看更详细的文档。

[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)配合其他[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)属性制作出来的实例：[demo](http://dev.opera.com/articles/view/beautiful-ui-styling-with-css3-text-shadow-box-shadow-and-border-radius/" \t "https://www.w3cplus.com/content/_blank)

那们今天我们有关[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的[box-shadow](http://www.w3.org/TR/2005/WD-css3-background-20050216/" \l "the-box-shadow" \t "https://www.w3cplus.com/content/_blank)就说到这里了，到今天为些我们一起探讨和学习了[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的[渐变Gradient](http://www.w3cplus.com/node/44" \t "https://www.w3cplus.com/content/_blank)、[透明色RGBA](http://www.w3cplus.com/node/45" \t "https://www.w3cplus.com/content/_blank)、[圆角border-radius](http://www.w3cplus.com/node/48" \t "https://www.w3cplus.com/content/_blank)和[文本阴影text-shadow](http://www.w3cplus.com/node/52" \t "https://www.w3cplus.com/content/_blank)，下一节我们将一起探讨[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的[transform](http://www.w3.org/TR/css3-2d-transforms/" \t "https://www.w3cplus.com/content/_blank)属性，感兴趣的朋友请观注本注有关[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)的博文更新情况，如果你有更好的学习想法，或者对本站介绍的[CSS3](http://www.w3.org/TR/css3-roadmap/" \t "https://www.w3cplus.com/content/_blank)有更好的建议可以随时联系我。

#### 更新一：

div {

filter:

progid:DXImageTransform.Microsoft.Shadow(color=#eeeeee,direction=0,strength=7)

progid:DXImageTransform.Microsoft.Shadow(color=#dddddd,direction=90,strength=10)

progid:DXImageTransform.Microsoft.Shadow(color=#dddddd,direction=180,strength=10)

progid:DXImageTransform.Microsoft.Shadow(color=#eeeeee,direction=270,strength=7);

}

使用滤镜来实现IE下的效果，基中“color”为阴影色，“direction”是阴影方向，“strength”是阴影强度。特别注意，颜色“#eeeeee”在此处不能写成“#eee”，不然会无效果。