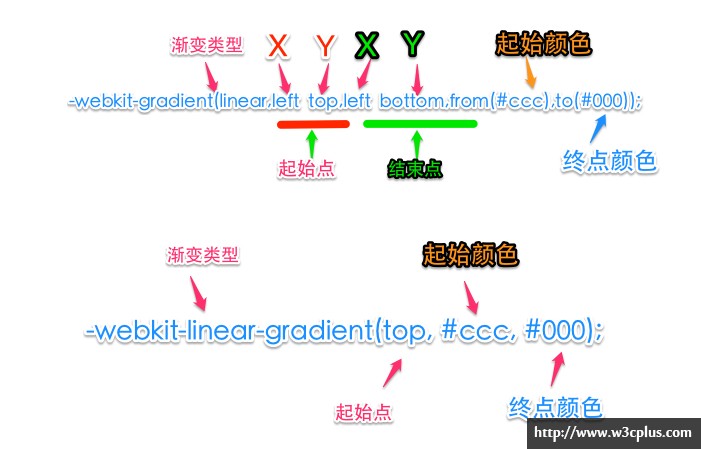
***线性渐变:***

***简化版本：***

**

**在此基础上的复杂渐变的写法：增加了中间颜色及所处的位置。**

**/\***

**老式写法：**

**background:-webkit-gradient(linear, left top, left bottom, from(#ff0000),color-stop(.5,#00ff00), to(#900)); \*/**

**注意：left top left 必须写在前面**

**新式写法：**

**background:-webkit-linear-gradient(top, #00f 50% ,#009 100%);**

**渐变的类型? (linear)**

**老式：**

**渐变开始的X Y 轴坐标(0 0 – 或者left-top)**

**渐变结束的X Y 轴坐标(0 100% 或者left-bottom)**

**新式：**

**（上述两句，在新型渐变标准中的写法变成了 top 或者left 或者度数 30deg）**

**老式：**

**开始的颜色? (from(red))**

**中间加入的颜色是中间色 color-stop(0.5 orange)**

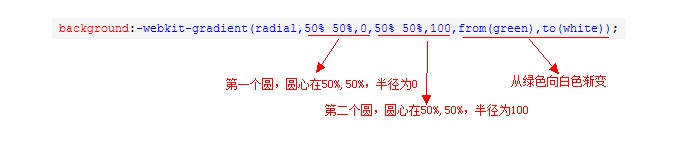
**结束的颜色? (to(blue))**

**新式的，前面是颜色，后面是百分比。**

**一、径向渐变的基础知识**

径向渐变的概念：从起点到终点颜色从内到外进行圆形渐变（从中间向外拉）。

**老写法：**



**新写法：前面是径向渐变的代表：radial 后面多了第三个属性 circle 和**ellipse 椭圆

background: -webkit-radial-gradient(left bottom , circle, #ace, #f96, #1E90FF);

**二、径向渐变的基本语法**

background-image:-webkit-gradient(type,x1 y1,r1,x2 y2,r2,from(开始颜色值),to(结束颜色值),color-stop(偏移量小数,停靠颜色值),... );

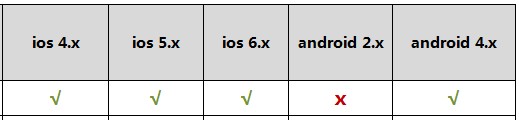
第一组参数type（类型）为 radial，因为是径向渐变；

第二组参数x1 y1,r1为第一个圆的圆心和半径；

第三组参数x2 y2,r2为第二个圆的圆心和半径；

第四组参数color-stop为渐变的中间过渡色，可以设置多组。

**二、CSS3径向渐变在ios和android系统的兼容测试**



android2.x特别是2.1等较低的版本，并不完全支持，渲染出来的效果不丰富

**二、CSS3径向渐变最简单的写法**

CSS3径向渐变（radial）的代码

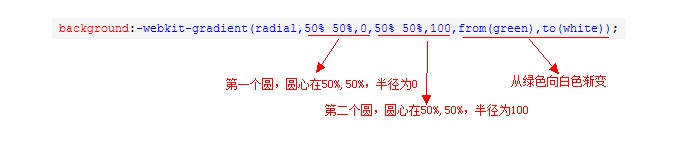
.bg{

　　background:-webkit-gradient(radial,50% 50%,0,50% 50%,100,from(green),to(white));

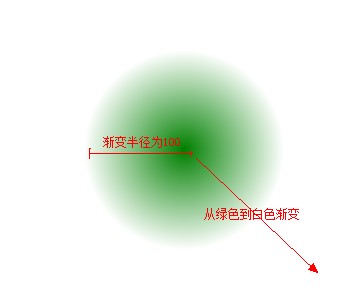
}

代码解释如下，可以看出径向渐变是2个圆组成，2个圆可分别设置不同的圆心和半径，这里的圆心都为(50%,50%)，第一个50%是圆心距离X轴的距离，第二个50%是圆心距离Y轴的距离，

而第一个圆的半径为0，第二个圆的渐变半径为100。



 运行代码后视觉效果如下图，可以看出以白色(white)结束后，白色填充了整个背景



大部分设计稿是一个颜色到另一个颜色的渐变，所以我们一般可以设置第一个圆渐变是起点，半径为0，第二个圆的渐变结束点，开始设置半径，这是最简单的做法啦！然后也可能运到更多复杂的效果，这个需要你对CSS3渐变有深入的理解和使用

CSS3 径向渐变和线性渐变是很相似的，我们首先来看其语法：

|  |  |
| --- | --- |
| 1  2 | -moz-radial-gradient([<bg-position> || <angle>,]? [<shape> || <size>,]? <color-stop>, <color-stop>[, <color-stop>]\*);   -webkit-radial-gradient([<bg-position> || <angle>,]? [<shape> || <size>,]? <color-stop>, <color-stop>[, <color-stop>]\*); |

　　除了您已经在线性渐变中看到的起始位置，方向，和颜色，径向梯度允许你指定渐变的形状（圆形或椭圆形）和大小（最近端，最近角，最远端，最远角，包含或覆盖 (closest-side, closest-corner, farthest-side, farthest-corner, contain or cover)）。 颜色起止(Color stops)：就像用线性渐变，你应该沿着渐变线定义渐变的起止颜色。下面为了更好的理解其具体的用法，我们主要通过不同的示例来对比CSS3径向渐变的具体用法

**示例一：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(#ace, #f96, #1E90FF);  background: -webkit-radial-gradient(#ace, #f96, #1E90FF); |

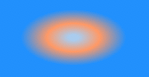
　　效果：



**示例二：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(#ace 5%, #f96 25%, #1E90FF 50%);  background: -webkit-radial-gradient(#ace 5%, #f96 25%, #1E90FF 50%); |

　　效果如下：



　　从以上俩个示例的代码中发现，他们起止色想同，但就是示例二定位了些数据，为什么会造成这么大的区别呢？其实在径向渐变中虽然具有相同的起止色，但是在没有设置位置时，其默认颜色为均匀间隔，这一点和我们前面的线性渐变是一样的，但是设置了渐变位置就会按照渐变位置去渐变，这就是我们示例一和示例的区别之处：虽然圆具有相同的起止颜色，但在示例一为默认的颜色间隔均匀的渐变，而示例二每种颜色都有特定的位置。

**示例三：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(bottom left, circle, #ace, #f96, #1E90FF);  background: -webkit-radial-gradient(bottom left, circle, #ace, #f96, #1E90FF); |

　　效果如下：



**示例四：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(bottom left, ellipse, #ace, #f96, #1E90FF);  background: -webkit-radial-gradient(bottom left, ellipse, #ace, #f96, #1E90FF); |

　　效果如下：

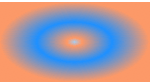


　　示例三和示例四我们从效果中就可以看出，其形状不一样，示例三程圆形而示例四程椭圆形状，也是就是说他们存在形状上的差异。然而我们在回到两个示例的代码中，显然在示例三中设置其形状为 circle，而在示例四中 ellipse，换而言之在径向渐变中，我们是可以设置其形状的。

**示例五：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(ellipse closest-side, #ace, #f96 10%, #1E90FF 50%, #f96);  background: -webkit-radial-gradient(ellipse closest-side, #ace, #f96 10%, #1E90FF 50%, #f96); |

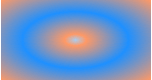
　　效果如下：



**示例六：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(ellipse farthest-corner, #ace, #f96 10%, #1E90FF 50%, #f96);  background: -webkit-radial-gradient(ellipse farthest-corner, #ace, #f96 10%, #1E90FF 50%, #f96); |

　　效果如下：

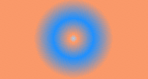


　　从示例五和示例六中的代码中我们可以清楚知道，在示例五中我人应用了closest-side而在示例六中我们应用了farthest-corner。这样我们可以知道在径向渐变中我们还可以为其设置大小(Size)：size的不同选项(closest-side, closest-corner, farthest-side, farthest-corner, contain or cover)指向被用来定义圆或椭圆大小的点。 示例：椭圆的近边VS远角 下面的两个椭圆有不同的大小。示例五是由从起始点(center)到近边的距离设定的，而示例六是由从起始点到远角的的距离决定的。

**示例七：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(circle closest-side, #ace, #f96 10%, #1E90FF 50%, #f96);  background: -webkit-radial-gradient(circle closest-side, #ace, #f96 10%, #1E90FF 50%, #f96); |

　　效果如下：



**示例八：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(circle farthest-side, #ace, #f96 10%, #1E90FF 50%, #f96);  background: -webkit-radial-gradient(circle farthest-side, #ace, #f96 10%, #1E90FF 50%, #f96); |

　　效果如下：



　　示例七和示例八主要演示了圆的近边VS远边 ，示例七的圆的渐变大小由起始点(center)到近边的距离决定，而示例八的圆则有起始点到远边的距离决定。

**示例九：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(#ace, #f96, #1E90FF);  background: -webkit-radial-gradient(#ace, #f96, #1E90FF); |

　　效果如下：



**示例十：**

|  |  |
| --- | --- |
| 1  2 | background: -moz-radial-gradient(contain, #ace, #f96, #1E90FF);  background: -webkit-radial-gradient(contain, #ace, #f96, #1E90FF); |

　　效果如下：



　　示例九和示例十演示了包含圆 。在这里你可以看到示例九的默认圈，同一渐变版本，但是被包含的示例十的圆。

　　最后我们在来看两个实例一个是应用了中心定位和full sized,如下所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | /\* Firefox 3.6+ \*/   background: -moz-radial-gradient(circle, #ace, #f96);   /\* Safari 4-5, Chrome 1-9 \*/   /\* Can't specify a percentage size? Laaaaaame. \*/   background: -webkit-gradient(radial, center center, 0, center center, 460, from(#ace), to(#f96));   /\* Safari 5.1+, Chrome 10+ \*/   background: -webkit-radial-gradient(circle, #ace, #f96); |

　　效果如下：



　　下面这个实例应用的是Positioned, Sized，请看代码和效果：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | /\* Firefox 3.6+ \*/  /\* -moz-radial-gradient( [ || ,]? [ || ,]? , [, ]\* ) \*/  background: -moz-radial-gradient(80% 20%, closest-corner, #ace, #f96);  /\* Safari 4-5, Chrome 1-9 \*/  background: -webkit-gradient(radial, 80% 20%, 0, 80% 40%, 100, from(#ace), to(#f96));  /\* Safari 5.1+, Chrome 10+ \*/  background: -webkit-radial-gradient(80% 20%, closest-corner, #ace, #f96); |

　　效果如下：

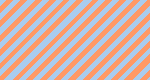


　　到此关于 CSS3 的两种渐变方式我们都介绍完了。再浪费大家一点时间，我们看看CSS3 **重复渐变**（Repeating Gradient）的应用**。**

　　如果您想重复一个渐变，您可以使用-moz-repeating-linear-gradient（重复线性渐变）和-moz-repeating-radial-gradient（重复径向渐变）。 在下面的例子，每个实例都指定了两个起止颜色，并无限重复。

|  |
| --- |
| background: -moz-repeating-radial-gradient(#ace, #ace 5px, #f96 5px, #f96 10px);  background: -webkit-repeating-radial-gradient(#ace, #ace 5px, #f96 5px, #f96 10px);  background: -moz-repeating-linear-gradient(top left -45deg, #ace, #ace 5px, #f96 5px, #f96 10px);  background: -webkit-repeating-linear-gradient(top left -45deg, #ace, #ace 5px, #f96 5px, #f96 10px); |

　　效果：

有关于CSS3渐变的东西就完了，大家看完了肯定会想，他主要用在哪些方面呢？这个说起来就多了，最简单的就是制作背景，我们还可以应用其制作一些漂亮的按钮，还可以用他来制作patterns，我在这里列出几种制作patterns的示例代码吧：

　　HTML代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <ul>     <li class="gradient gradient1"></li>     <li class="gradient gradient2"></li>     <li class="gradient gradient3"></li>     <li class="gradient gradient4"></li>     <li class="gradient gradient5"></li>     <li class="gradient gradient6"></li>  </ul> |

　　CSS 代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121 | ul {    overflow: hidden;    margin-top: 20px;  }  li{    width: 150px;    height: 80px;    margin-bottom: 10px;    float: left;    margin-right: 5px;    background: #ace;    /\*Controls the size\*/    -webkit-background-size: 20px 20px;    -moz-background-size: 20px 20px;    background-size: 20px 20px;  }    li.gradient1 {    background-image: -webkit-gradient(      linear,      0 100%, 100% 0,      color-stop(.25, rgba(255, 255, 255, .2)),      color-stop(.25, transparent),      color-stop(.5, transparent),      color-stop(.5, rgba(255, 255, 255, .2)),      color-stop(.75, rgba(255, 255, 255, .2)),      color-stop(.75, transparent),      to(transparent)      );    background-image: -moz-linear-gradient(      45deg,      rgba(255, 255, 255, .2) 25%,      transparent 25%,      transparent 50%,      rgba(255, 255, 255, .2) 50%,      rgba(255, 255, 255, .2) 75%,      transparent 75%,      transparent      );    background-image: -o-linear-gradient(      45deg,      rgba(255, 255, 255, .2) 25%,      transparent 25%,      transparent 50%,      rgba(255, 255, 255, .2) 50%,      rgba(255, 255, 255, .2) 75%,      transparent 75%,      transparent    );    background-image: linear-gradient(      45deg,      rgba(255, 255, 255, .2) 25%,      transparent 25%,      transparent 50%,      gba(255, 255, 255, .2) 50%,      rgba(255, 255, 255, .2) 75%,      transparent 75%,      transparent      );  }    li.gradient2 {     background-image: -webkit-gradient(linear, 0 0, 100% 100%,        color-stop(.25, rgba(255, 255, 255, .2)), color-stop(.25, transparent),        color-stop(.5, transparent), color-stop(.5, rgba(255, 255, 255, .2)),        color-stop(.75, rgba(255, 255, 255, .2)), color-stop(.75, transparent),        to(transparent));     background-image: -moz-linear-gradient(-45deg, rgba(255, 255, 255, .2) 25%, transparent 25%,        transparent 50%, rgba(255, 255, 255, .2) 50%, rgba(255, 255, 255, .2) 75%,        transparent 75%, transparent);     background-image: -o-linear-gradient(-45deg, rgba(255, 255, 255, .2) 25%, transparent 25%,        transparent 50%, rgba(255, 255, 255, .2) 50%, rgba(255, 255, 255, .2) 75%,        transparent 75%, transparent);     background-image: linear-gradient(-45deg, rgba(255, 255, 255, .2) 25%, transparent 25%,        transparent 50%, rgba(255, 255, 255, .2) 50%, rgba(255, 255, 255, .2) 75%,        transparent 75%, transparent);  }    li.gradient3 {    background-image: -webkit-gradient(linear, 0 0, 0 100%, color-stop(.5, rgba(255, 255, 255, .2)), color-stop(.5, transparent), to(transparent));    background-image: -moz-linear-gradient(rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);    background-image: -o-linear-gradient(rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);    background-image: linear-gradient(rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);  }    li.gradient4 {    background-image: -webkit-gradient(linear, 0 0, 100% 0, color-stop(.5, rgba(255, 255, 255, .2)), color-stop(.5, transparent), to(transparent));    background-image: -moz-linear-gradient(0deg, rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);    background-image: -o-linear-gradient(0deg, rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);    background-image: linear-gradient(0deg, rgba(255, 255, 255, .2) 50%, transparent 50%, transparent);  }    li.gradient5 {    background-image: -webkit-gradient(linear, 0 0, 100% 100%, color-stop(.25, #555), color-stop(.25, transparent), to(transparent)),        -webkit-gradient(linear, 0 100%, 100% 0, color-stop(.25, #555), color-stop(.25, transparent), to(transparent)),        -webkit-gradient(linear, 0 0, 100% 100%, color-stop(.75, transparent), color-stop(.75, #555)),        -webkit-gradient(linear, 0 100%, 100% 0, color-stop(.75, transparent), color-stop(.75, #555));    background-image: -moz-linear-gradient(45deg, #555 25%, transparent 25%, transparent),       -moz-linear-gradient(-45deg, #555 25%, transparent 25%, transparent),       -moz-linear-gradient(45deg, transparent 75%, #555 75%),       -moz-linear-gradient(-45deg, transparent 75%, #555 75%);    background-image: -o-linear-gradient(45deg, #555 25%, transparent 25%, transparent),       -o-linear-gradient(-45deg, #555 25%, transparent 25%, transparent),       -o-linear-gradient(45deg, transparent 75%, #555 75%),       -o-linear-gradient(-45deg, transparent 75%, #555 75%);    background-image: linear-gradient(45deg, #555 25%, transparent 25%, transparent),      linear-gradient(-45deg, #555 25%, transparent 25%, transparent),      linear-gradient(45deg, transparent 75%, #555 75%),      linear-gradient(-45deg, transparent 75%, #555 75%);  }    li.gradient6 {    background-image: -webkit-gradient(linear, 0 0, 0 100%, color-stop(.5, transparent), color-stop(.5, rgba(200, 0, 0, .5)), to(rgba(200, 0, 0, .5))),       -webkit-gradient(linear, 0 0, 100% 0, color-stop(.5, transparent), color-stop(.5, rgba(200, 0, 0, .5)), to(rgba(200, 0, 0, .5)));    background-image: -moz-linear-gradient(transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5)),       -moz-linear-gradient(0deg, transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5));    background-image: -o-linear-gradient(transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5)),       -o-linear-gradient(0deg, transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5));    background-image: linear-gradient(transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5)),       linear-gradient(0deg, transparent 50%, rgba(200, 0, 0, .5) 50%, rgba(200, 0, 0, .5));  } |

　　效果：

http://www.w3cplus.com/sites/default/files/gradient29.png

　　不错的效果吧，当然感兴趣的朋友可以到[这里](http://leaverou.me/css3patterns/)学习制作更多的不同效果。