## 背景

### 给背景设置和去除边框

默认情况下背景会延伸到边框所在区域的下层，比如给一个有背景的元素应用一道老土的虚线边框，就可以看出来。即使你使用的是不透明的实色边框，这个事实也不会有任何改变。在css2.1中，这就是背景的工作原理。图如下：



从css3开始可以采用background-clip属性来调整默认行为带来的不便，即可以设置边框。此属性有3个值：

1. border-box：背景会被元素边框外沿框裁切掉
2. content-box：背景会被裁切到内容框
3. Padding-box：背景会被裁切到内边距的外沿

例子：

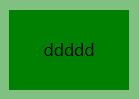




将background-clip的值设置为padding-box或者content-box，产生如下效果(边距还是实际存在)：

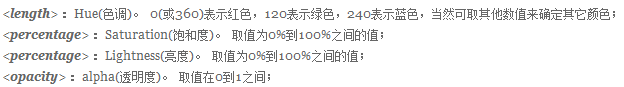


设置为border-box：



例子中hasl函数介绍：

语法：



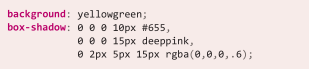
### 设置多重边框

可以通过box-shadow属性来设置多重边框和投影，其语法格式如下：

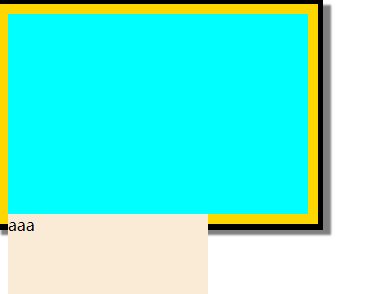


其中第四个参数可以通过指定正值或负值，可以让投影面积加大或者减小。

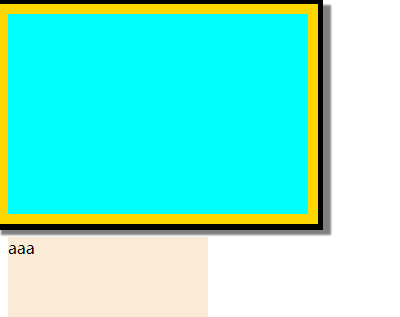
box-shadow 是层层叠加的，第一层投影位于最顶层，依次类推。因此，你需要按此规律调整扩张半径。比如说，在前面的代码中，我们想在外圈再加一道 5px 的外框，那就需要指定扩张半径的值为15px （ 10px+5px ）。如果你愿意，甚至还可以在这些“边框”的底下再加一层常规的投影：



通过此属性设置的边框或者投影，它不会影响到布局，也不会受到box-sizing属性的影响不过，如下所示设置了两个边距和一个投影，一共三个边距，但最终是没有影响到布局的：

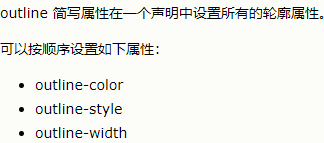


你还是可以通过内边距或外边距（这取决于投影是内嵌和还是外扩的）来额外模拟出边框所需要占据的空间，如下为设置了具体的边距：



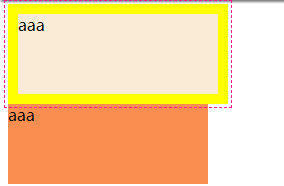
上述方法所创建出的假“边框”出现在元素的外圈。它们并不会响应鼠标事件，比如悬停或点击。如果这一点非常重要，你可以给box-shadow 属性加上 inset 关键字，来使投影绘制在元素的内圈。请注意，此时你需要增加额外的内边距来腾出足够的空隙。

在某些情况下，你可能只需要两层边框，那就可以先设置一层常规边框，再加上outline（描边）属性来产生外层的边框。这种方法的一大优点在于边框样式十分灵活，不像上面的 box-shadow 方案只能模拟实线边框（假设我们需要产生虚线边框效果， box-shadow 就没辙了）。outline属性的语法如下：



还可以使用outline-offset属性来控制外层边框和元素之间的距离，此属性还可以设置负值。

outline和outline-offset的例子：



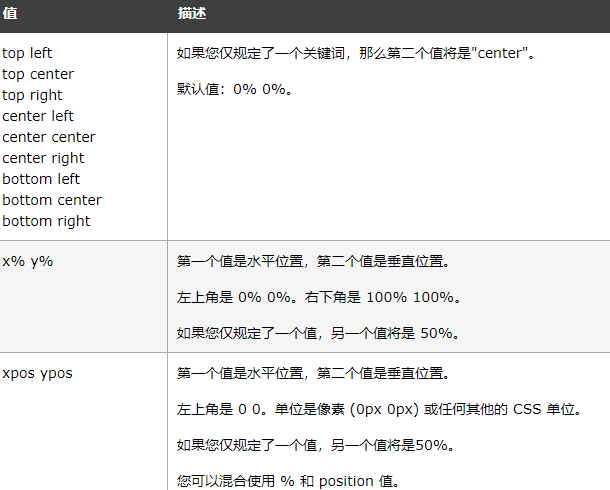
#### 圆角边框

border-radius

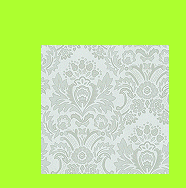
### 背景定位

**background-position**

语法格式如下：

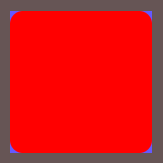




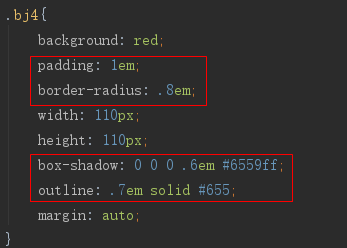


### 边框内圆角

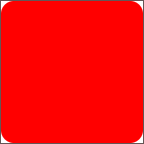
实现如图所示的效果：



关键css代码：



上面得效果主要使用了圆角和描边，圆角是由border-radius来控制的，描边是由box-shadow和outline来控制的。如果不加box-shadow，产生的效果会像下面一样：



通过找个图可以得知，产生那种描边的效果是通过设置box-shadow来填充达到的。主要是设置合适的描边宽度，真好填满图中的空白。关于如何设置，这里有一个定理：

扩展半径需要比描边的宽度值小，但同时又要比(这里的r表示border-radius）。这意味着如果描边的宽度比，那就不可能使用此方式达成描边填充效果的。

### 条纹背景（线性渐变）

#### 横向背景

产生横向的条纹背景：background: linear-gradient(颜色,..)

如：background:linear-gradient(blue,yellow,red,black,green);

具体查看：《css揭秘》---第二章 背景与边框的垂直条纹

#### 竖向背景

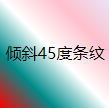


效果如下：



#### 指定倾斜角度的背景





linear-gradient() 和 radial-gradient() 还各有一个循环式的加强版： repeating-linear-gradient() 和 repeating-radial-gradient() 。它们的工作方式跟前两者类似，只有一点不同：色标是无限循环重复的，直到填满整个背景，使用它们可以随心所欲的改变渐变的角度，如下所示：









如果需要产生垂直或竖向条纹，最好是使用上面的linear-gradient

#### 灵活的同色系条纹

根据 CSS 图像（第四版）（http://w3.org/TR/css4-images）计划新增的一个简化语法来看，很快我们就可以在同一个色标上指定两个位置值了。这个简写语法的含义相当于两个连续的色标具有相同的颜色和不同的位置，这个特性在创建渐变图案时是十分有用的。



### 复杂的背景图案

[https://philiprogers.com/svgpatterns/#microbialmat](https://philiprogers.com/svgpatterns/" \l "microbialmat)