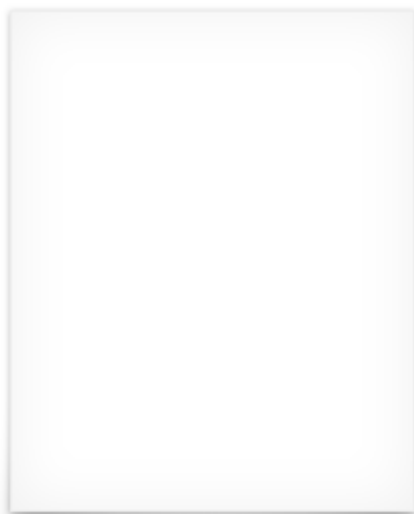


## 详解 box-shadow

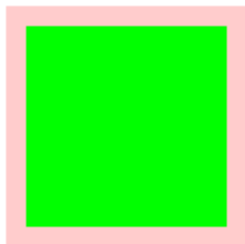
今天我们来探究一下“阴影”。这个阴影可不是什么心理阴影。我们探究的阴影是图形学中的阴影。同学们可以先想想，在图片中有了阴影，或者使用了阴影之后，为我们的图像提供了哪些特殊的效果？在高中大家在学习美术的时候，那个时候接触素描，去画一个杯子，或者画一个球体，这个时候往往出现，老师画的很逼真，**立体感、层次感**很强，自己虽说画出来了，在不刻意强调的前提下容易让人们误会（滑稽脸）。经过老师的讲解，我们知道想让我们所画的物体，具有立体感，通常有：**高光，暗面，灰面，投影（阴影）**，这几部分组成。说完了图形中的阴影后，来想想如何在我们的页面中绘画出阴影。这就涉及到我们今天要说的内容 CSS3 中的 box-shadow。

### 探究 box-shadow

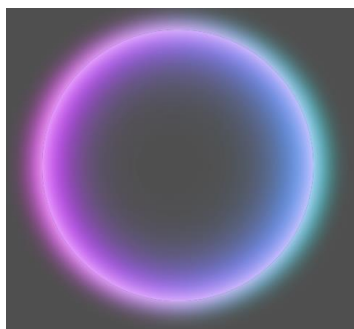
在先说 box-shadow 之前，大家先来看看几张图片，具体细致的感受一下 box-shadow 阴影。



一张很像纸的纸（废话），有点阴影。



What? 这也是阴影？（不要拿 border 来骗我好吧。。。）



这个也用 `box-shadow` 画出来的？

上面的这些图片确实是通过 `box-shadow` 画出来的，同学们可能有些小震惊（图一很微妙，图三很炫酷，图二有点开玩笑）。接下来我们来正式探究 `box-shadow` 的神奇黑魔法！！！！。

`box-shadow` 直译被称为“盒子阴影”，直译的结果与同学们的认知是相符的。CSS 中存在盒模型,针对于盒模型的阴影。这点大家要记住，是针对于盒子（元素）的阴影，后面我们还会着重说明。

`box-shadow` 属性值由 6 部分组成。

语法形式：

**`box-shadow: offset-x offset-y blur spread color position;`**

这里的 `position` 可以写在最前面。也可以像上面一样写在最后面。

接下来我们来看看这几个属性都提供了什么样的功能？

#### 1. `offset-x offset-y`

`offset-x` 用于声明阴影的水平偏移，就是阴影在 `x` 轴上的位置。当值为正数，阴影就位于元素右侧，若值为负数，阴影位于元素的左侧。

同理

`offset-y` 用于声明阴影的垂直偏移，就是阴影在 `y` 轴上的位置。当值为正数，阴影就位于元素下侧，若值为负数，阴影位于元素的上侧。

例子：

CSS:

```
.demo {  
  width: 100px;  
  height: 100px;  
  background: #fcc;  
  box-shadow: 10px 10px;  
}
```

结果：



通过这个我们能看出来是具有阴影的，根据上面的讲解能看出来是（10px 10px）向下向右偏移 10 个 px，同理负值是向上向左偏移。接下来我们看 blur

## 2. blur

blur 表示阴影的模糊半径。效果与设计软件中使用的高斯模糊滤镜一样。值为 0 意味着阴影完全不模糊。blur 值越大，边角越不锋利，阴影越朦胧。不允许负值，负值等同于 0。在我们上面的例子中修改：

CSS:

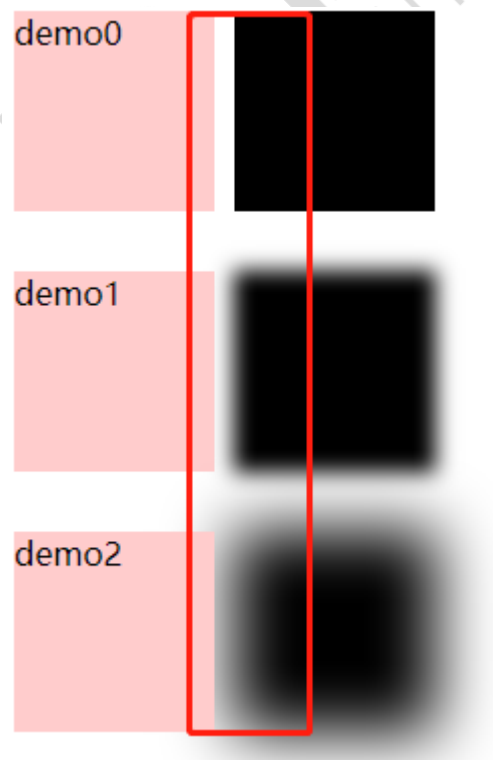
```
box-shadow: 10px 10px 10px;
```

结果：



我们能看到阴影虚化了。

这里要注意的是：W3C 中没有规定浏览器厂商使用哪种方式实现模糊效果，反正效果与高斯模糊效果差不多就是了。但有一点我们需要注意的，那就是模糊效果会扩大阴影的面积。



Demo0 是 blur 半径为 0 的效果，可以看到阴影尺寸与元素尺寸一模一样。而 demo1 是 blur 半径为 10px 的效果，可以看到阴影尺寸有所扩展了，而 demo2 则扩展得更多一些。

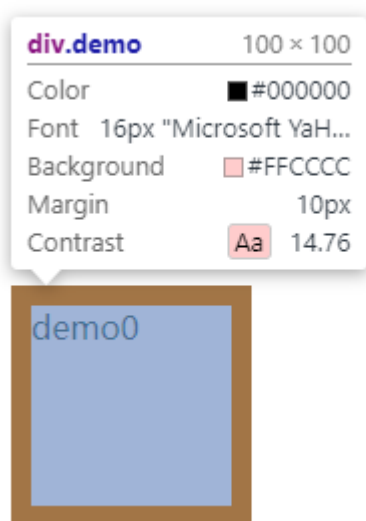
现在我们感性上认知到 `blur` 半径值大于 0 时会扩展阴影尺寸，那么到底扩展多少呢？那我们要先明确模糊发生的起始位置了。经过肉眼观察，模糊发生的起始位置就是阴影盒子的各边。事实也就是这样。至于发散距离

`blur radius(虚化半径) / 2` 的距离。看的 demo2 中阴影尺寸已经与元素盒子重叠了，因为阴影盒子左边框向左发散了 15px 了，超过它俩之间 10px 的水平距离了。

### 3. spread

`spread` 表示阴影的大小。当值为正数，阴影会向四周扩展。若值为负数，阴影会收缩，小于元素尺寸。默认值 0 会保持阴影和元素尺寸一致。

`box-shadow: 0px 0px 0px 10px;`



在这里面我们能看到，当我们 `spread` 为 10px，`margin` 也为 10px 他俩是重合的，说明 `spread` 为正值，在原来的大小上（上下左右）+ `spread` 的值。当 `spread` 为为负值，再远的大小上 - `spread` 的值

### 4. color

`color` 表示阴影的颜色。可以是任何颜色单位。这个没什么说的。

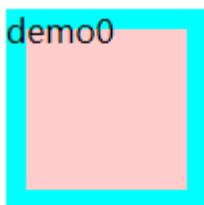
### 5. position

`position` 表示阴影的位置，可选项。默认为外部阴影。可以通过使用 `inset` 关键字来制作内部阴影。外部阴影不用声明默认就是，想使用内部阴影需要声明 `inset` 在最前面最后都可以。

Like this:

`box-shadow: inset 0px 0px 0px 10px #0ff;`

效果:

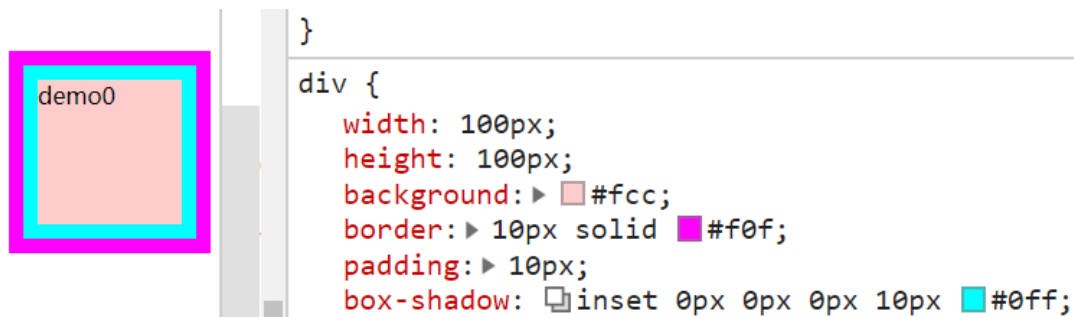


很明显阴影是在里面的。

接下来我们也对内阴影分析一下 `spread` 以及 `blur radius` 。

通过上面的图我们能看出来。

对于 `spread` 当值为正值时，可以看出内阴影从边界开始填充。至于这个边界是从 `border` 开始，还是从 `padding` 开始呢？



CSS:

```

border: 10px solid #f0f;
padding: 10px;
box-shadow: inset 0px 0px 0px 10px #0ff;

```

我们可以看出来

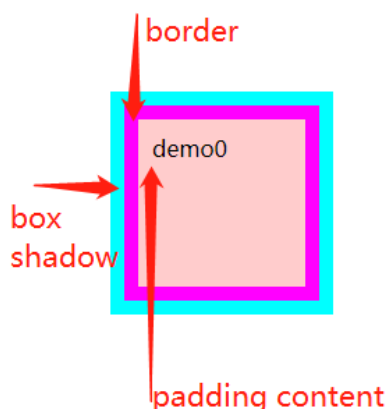
以 `Padding` 区域为起点进行向内部扩展阴影。扩展半径为 `spread` 的值，负值没有意义，没有 `padding` 区域以 `content` 区域开始。

对于 `blur radius` 和 `spread` 类似这里就不演示了，给大家总结一下，模糊半径的值还是 `blur radius/2` 开始位置同 `spread`，有 `padding` 先从 `padding` 开始，没有从 `content` 开始。

下面说一说 `box-shadow` 的层次（`z-index`）

对于

外阴影（默认情况）：



对于外阴影：高与 `margin` 低于 `background`。

对于内阴影：高与 `padding` 低于 `content`。

`Box-shadow` 不仅可以区分内外阴影还可以，应用多个阴影进行叠加。

每个阴影用 ‘,’ 隔开，like this

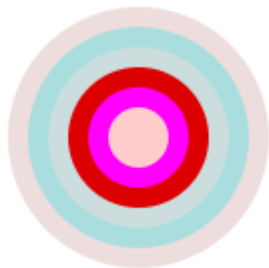
CSS:

```
width: 100px;
height: 100px;
background: #fcc;
padding: 10px;
border-radius: 50%;
box-shadow: 120px 0px 0 -10px #f0f,
95px 10px 0 0px #d00,
30px 20px 0 -10px #cdd,
90px -10px 0 0px #add,
40px -30px 0 0px #edd;
```

结果：



层次也是后写的在后边。因为可以重叠所以我们可以写多边框的一个元素



这样的一个准靶。（这个也就是刚开始出现的那张图片的原因）。