

border-radius 你了解多少

在 `border-radius` 这个 CSS3 属性出现之前，我们要想在页面画个圆，务必借助图片来实现，出了这个 `border-radius` 属性之后，我们在页面中实现一个圆形是非常容易的。接下来我们仔细研究研究这个 `border-radius`，看看它还有那些黑魔法？

Border-radius 探究

通过 `border-radius` 轻而易举的可以划出一个圆。我们通过这样的一个小例子，开始探究 `border-radius`。

HTML:

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

CSS:

```
.demo {  
    width: 100px;  
    height: 100px;  
    background-color: #f00;  
    border-radius: 50px;  
}
```

结果:



要想画正圆，我们首先要有一个正方形，所以在宽高 100px 的正方形，加上 `border-radius: 50px`；同学们可能好奇，它是怎么计算。50px 代表的是什么？这时我们看看控制台：

```
.demo {  
    width: 100px;  
    height: 100px;  
    background-color: #f00;  
    border-radius: 50px;  
        border-top-left-radius: 50px;  
        border-top-right-radius: 50px;  
        border-bottom-right-radius: 50px;  
        border-bottom-left-radius: 50px;  
}
```

上面的内容使我们展开的样子，虽说我们仅仅写了一个值 50px；但是对于浏览器来说，他认为是四个一样的值。对于 `border-*-radius` 这种展开属性一一对应。看到这里大家能想到的类似的属性（设置一个属性，会被应用到多个展开属性的属性）：

`margin(margin-left, margin-top, margin-right, margin-bottom)`

`padding(padding-left, padding-top, padding-right, padding-bottom)`

`border(border-width, border-style, border-color, 还可以继续展开)`

对于 `margin` 和 `padding` 来说，大家都知道他们的顺序是上右下左。以及 1 个值，多个值对应是什么，在这里就不多说了。类比的来看 `border-radius` 他的顺序，根据属性来看，不是上下左右这么简单。而是：左上，右上，右下，左下。

根据上面的分析，我们要是写：

`border-radius: 50px;` 与 `border-radius: 50px 50px 50px 50px;` 一致。

（类似 `margin: 10px;` 与 `margin: 10px 10px 10px 10px;`）

与：

```
border-top-left-radius: 50px;
border-top-right-radius: 50px;
border-bottom-right-radius: 50px;
border-bottom-left-radius: 50px;
```

这里同学们要注意：我们习惯说，左上，右上，但是这个属性表示的时候先说明上下，再说左右，也就是 `border-top-left-radius`。

从上面的四个属性可以看出我们的圆，应该是由四部分构成，我们修改一下参数会发生什么？

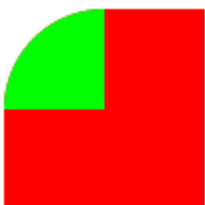
CSS:

```
.demo {
    width: 100px;
    height: 100px;
    background-color: #f00;
    border-radius: 50px 0 0 0;
}
```

结果：



我们这么写，相当于 `border-top-left-radius: 50px;` 再看一张图：



通过这张图能说明白我们看到的这个圆角，其实是：以左上角为起点，以水平

50px，垂直 50px 的点为圆心，50px 为半径画的圆。与原本 div 的左上角的重叠部分。就是我们看到的圆角。同理：

Border-radius: 0 50px 0 0; 与 border-top-right-radius: 50px;



这个是：以右上角为起点，以水平 50px，垂直 50px 的点为圆心，50px 为半径画的圆。与原本 div 的右上角的重叠部分。

同理右下，和左下我就不解释了。就是起点位置时右下角和左下角。同样这里需要注意。我们的圆不是直接画出来的，不是一个圆心，是四个圆心画出的四个圆与原来的 div 对应方向重叠形成的圆。

接下来探究一下，我们的属性能不能为负值？最大值又是多少？分别是什么含义？

1. 能为负值么？

我们设置成负值，看一下。

border-radius: -10px;

样式就不看，我们看一下样式报错：

⚠ border-radius: -10px;

说明值是无效的。为什么是无效的呢，上面我们都分析过了，我们要根据数值进行画圆。但是圆的半径是没有负值的。就算是可以为负值那就画到原来 div 的外侧了没有实际意义。所以不会有负值。

2. 最大值是多少？

按照我们上面的分析，当我们设置的半径越大，圆也会变化，是不是意味着圆角可以无限大呢？我们试一下（假设宽高为一百像素的元素）

50px(50%):

border-top-left-radius: 50px;

结果：



100px(100%):

border-top-left-radius: 100px;

结果：



>100px(>100%):

`border-top-left-radius: 150px;`

结果:



发现没有变，这说明：**圆的半径最大为 div 宽高的的一倍。** 长方形咋办？我们来试一下：

```
.demo {
  width: 200px;
  height: 100px;
  background-color: #f00;
  border-top-left-radius: 200px;
}
```

这是一个宽 200px 高 100px 的长方形。我们圆角的半径 200px，是宽的一倍，是高的两倍。根据上面我们的分析，应该是一个宽 200 高 100 的椭圆。我们看一下结果：



不难发现是一个宽高 100 像素的圆。这跟我们的预期不一样啊。

这里还有一个知识点要说明一下：

`Border-top-left-raidus: x y;`

看到这里大家似乎明白了，我们可以分别设置 x 偏移和 y 偏移，那意味着我们可以真正做椭圆了，我们试一下。

`border-top-left-radius: 200px 100px;`

结果:



一看还真是椭圆。

上面的那种情况

`border-top-left-radius: 100px;`

与

`border-top-left-radius: 200px;`

表现形式一致。

这个总结一下：

当使用一个值的时候，宽和高一致，当任何一个值（高和宽）超过宽高的最小值。宽高默认选择最短的值，也就是都选小值。因为 200 不能又表示 200，又表示 100 只能 100 100。

当使用两个值的时候，情况有些不一样了。看下面的例子：

```
.demo {
  width: 100px;
  height: 100px;
  background-color: #f00;
  border-top-left-radius: 2000px 100px;
}
```

你没有看错就是 2000px；

我们看一下结果。

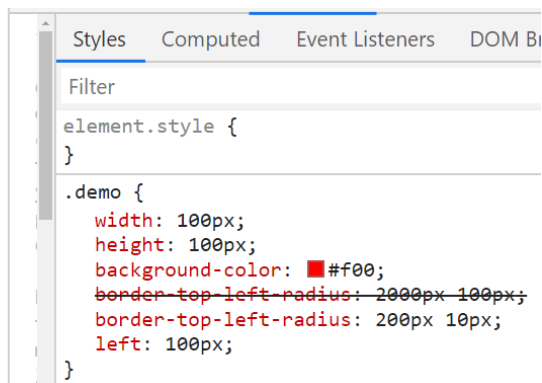


这里呢，我就直接说结果了。

当分开写时，最大值超过宽高最大值，会按照，宽高最大值等比缩放。

2000: 100

那我们看看 200: 10



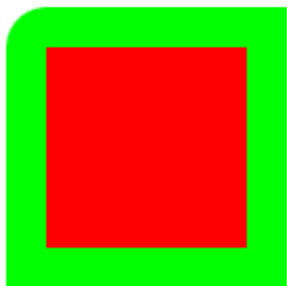
其实是一样的。（这里可能出现的情况，当图片过大 GPU 渲染会发生偏差，可能不一样）。

接下来我们看看，如果有 border，这个圆角又该怎么算？
我们设置一个宽高 100px，边框 20px，圆角 20px

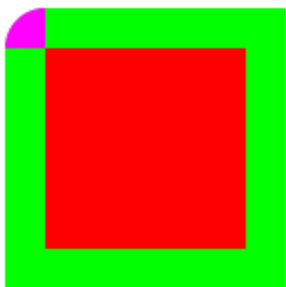
CSS:

```
.demo {
    width: 100px;
    height: 100px;
    border: 20px solid lime;
    background-color: #f00;
    border-radius: 20px 0 0 0;
}
```

结果:



我们看出来外边框也被画成的圆角。是怎么画出来的呢，我借助一张图辅助解释一下：



在原来的基础画了一个 border-top-left-radius 为 20px。宽高 20px 的正方扇形，正好把圆角盖住。

起点仍是左上角，画圆规则没变，最大值为

(border-top/border-left 最小值)

+

(borderd-right/border-bottom 最小值)

+

(padding-top,padding-left 最小值)

+

(padding-bottom,padding-right 最小值)

+

width/height 最小值

像这样：

```
.demo {
    width: 100px;
    height: 50px;
    padding: 20px;
    border: 20px solid lime;
    background-color: #f00;
    border-top-left-radius: 130px;
}
```

$20 + 20 + 20 + 20 + 50 = 130$

对于以上公式的计算，大家了解一下就可以。当我们属性分开写时。涉及到等比缩放，这里就不探究了，同学们可以自行尝试。

最后说一下 border-radius 高级写法

我们可以明确声明每个角的 x,y 同学们的第一个反应是

```
border-top-left-radius: 10px 20px;
border-top-right-radius: 10px 20px;
border-bottom-right-radius: 10px 20px;
border-bottom-left-radius: 10px 20px;
```

这里有一种更简单的表达形式：

```
border-radius: 10px / 20px;
```

```
border-radius: 10px 10px 10px 10px / 20px 20px 20px 20px;
```

等同于上面的写法，就是把 x 和 y 的值，分别写出来，用 / 分开。

最后在补充一小点。

我们 border-radius: 后面两个值，和三个值的含义。

两个值：

```
border-radius: ▼ 10px 20px;
border-top-left-radius: 10px;
border-top-right-radius: 20px;
border-bottom-right-radius: 10px;
border-bottom-left-radius: 20px;
```

当 `border-radius` 参数为两个值时 (x, y) 代表含义是

左上, 右下: x

右上, 左下: y

三个值:

```
border-radius: 10px 20px 30px;  
border-top-left-radius: 10px;  
border-top-right-radius: 20px;  
border-bottom-right-radius: 30px;  
border-bottom-left-radius: 20px;
```

当 `border-radius` 参数为三个值时 (x, y, z) 代表含义是:

左上: x

右上, 左下: y

右下: z