



CSS3 border 边框

讲师：许井龙

微信: ngsteel

- 一. CSS3 新增的边框样式**
- 二. CSS3 边框颜色**
- 三. CSS3 图片边框**
- 四. CSS3 圆角边框**
- 五. CSS3 盒子阴影**
- 六. CSS3 元素外轮廓**
- 七. CSS3 调整元素盒子大小**

一、CSS3 边框简介

- CSS1 与 CSS2 的边框属性
 - border-width, 设置元素四个边框的宽度
 - border-color, 设置元素四个边框的颜色
 - border-style, 元素四个边框的样式。详细参考: [mozilla开发者中心](#)
 - 上述三个属性都遵循 “**TRBL原则**” (上, 右, 下, 左)。
 - 简写方式 border: *border-width, border-color, border-style*

- CSS3 新增的边框样式

- `border-(top | right | bottom | left)-color`: 分别设置元素四个边框的颜色。由于浏览器兼容性问题，该属性极少在商业网站中使用。
- `border-image`: 使用图片作为元素的边框。浏览器支持较好，但商业网站应用很少，主要用在个人博客中。
- `border-radius`: 让元素显示圆角效果。浏览器支持较好，应用非常广泛。
- `box-shadow`: 让元素显示阴影效果。应用非常广泛，主要用来呈现元素立体效果。

- CSS3边框 语法：
 - border-top-color: [<color>], 上边框颜色
 - border-right-color: [<color>], 右边框颜色
 - border-bottom-color: [<color>], 下边框颜色
 - border-left-color: [<color>], 左边框颜色
 - 上述四个样式属性, 仍然遵循TRBL原则
 - 简写方式, border-color: [<color> | transparent] {1, 4} | inherit,

特别说明:

- 1、该样式可以应用到所有的元素, 甚至伪元素。例如::first-letter。
- 2、该样式具有动画效果。

- 为边框设置多个颜色（仅Firefox浏览器支持）
 - `-moz-border-top-colors: [color]{1,n}`
 - `-moz-border-right-colors: [color]{1,n}`
 - `-moz-border-bottom-colors: [color]{1,n}`
 - `-moz-border-left-colors: [color]{1,n}`
 - 每个属性后面可以设置N个合法颜色值，每个颜色占1px。如果元素的边框设置了20px，而颜色只设置了10个，剩下的10px都使用最后一个颜色值。
- 注意：
- 上述属性仅在Firefox3.0以上有效，它不是标准语法。而且目前仍在实验阶段，实际使用一定要慎重。
 - 上述样式的优先级比 `border-[top | right | bottom | left]-color` 优先级高。
 - 不支持动画效果

示例代码：立体效果的边框

```
-moz-border-bottom-colors: #000 #111 #222 #333 #444 #555 #666 #777 #888 #999 #aaa #bbb #ccc #ddd #eee #fff;  
-moz-border-top-colors: #000 #111 #222 #333 #444 #555 #666 #777 #888 #999 #aaa #bbb #ccc #ddd #eee #fff;  
-moz-border-left-colors: #000 #111 #222 #333 #444 #555 #666 #777 #888 #999 #aaa #bbb #ccc #ddd #eee #fff;  
-moz-border-right-colors: #000 #111 #222 #333 #444 #555 #666 #777 #888 #999 #aaa #bbb #ccc #ddd #eee #fff;
```

需要配合

`border: 10px solid;`
一起使用。

注意：该样式仅在Firefox浏览器下可使用

图片边框

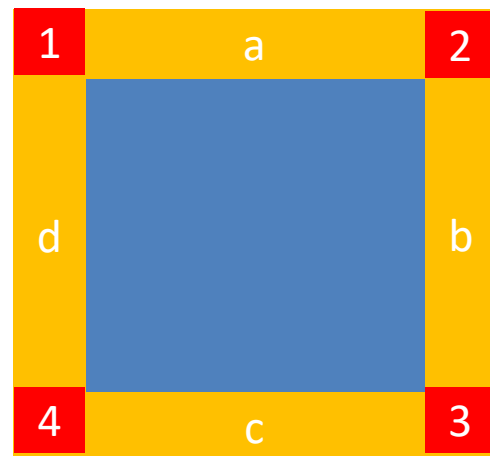
boder-image

- border-image-source: url(图片路径)
 - 设置图片路径。

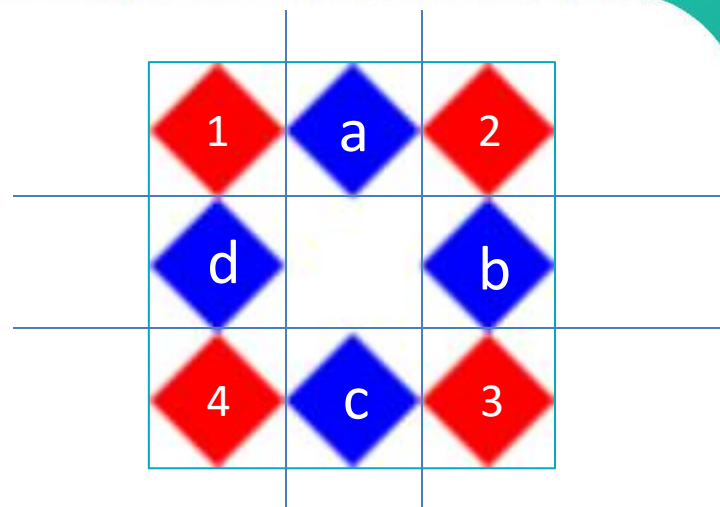
图片边框样式生效的前提: 必须设置 border 的宽度和 样式

例如, border: 1px solid 或者 border: 2px dotted 等等

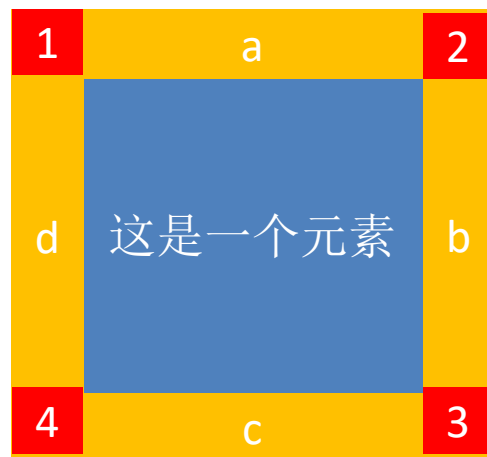
- border-image-slice: [<percentage> | <number>] {1,4} && fill。
- 图片切割的方式。让元素的边框的 **四个角**，**四条边** 显示切割后的图片。
- 可选值
 1. 关键字: fill (默认值) **图片填充边框的四个角。**
 2. 数字{1, 4} (不需要写单位, 默认为px)
 3. 百分比值 {1, 4} --- 基于图片的宽度或者高度



- border-image-slice:
- 使用数值
 - 26
 - 等价于 26 26 26 26
- 切割顺序及位置
 1. 距离图片上边 26px 位置切割
 2. 距离图片右边 26px 位置切割
 3. 距离图片下边 26px 位置切割
 4. 距离图片左边 26px 位置切割
- 经过上述切割
 - 我们已经把一张图片切割成了9份
 - 按照其编号分别对应元素边框的4个角、4个边（中间的图不需要）



图片原始大小
78px * 78px



- border-image-width: [<length>] {1,4}
 - 用来表示背景图片显示的大小。
 - 与border-width设置效果相同，**但是不会导致元素的实际大小改变。**
- 样式值
 - 10px 四个图片边框都是10px
 - 10px 20px 上下边框 10px , 左右边框 20px
 - 10px 20px 30px 上边框 10px ,左右边框 20px, 下边框 30px
 - 10px 20px 30px 40px 上边框 10px ,右边框 20px, 下边框 30px, 左边框 40px

- border-image-repeat: [stretch | round | repeat]
 - stretch: 默认值 四个边个图片被拉伸
 - round: 填满 四个边的图片重复平铺，**确保图片完整性。**
 - repeat: 重复 四个边的图片重复平铺

- `/* border-image-width: all */`
 - `border-image-width: 3;`
- `/* border-image-width: vertical horizontal */`
 - `border-image-width: 2em 3em;`
- `/* border-image-width: top horizontal bottom */`
 - `border-image-width: 5% 15% 10%;`
- `/* border-image-width: top right bottom left */`
 - `border-image-width: 5% 2em 10% auto;`

- border-image-outset: [<length>] {1,4}
- 图片外边框的位置，只能为整数，位移不会改变盒子的大小。
- 样式值
 - 10px 四个图片边框都向外移动10px
 - 10px 20px 上下边框向外移动10px，左右边框向外移动20px
 - 10px 20px 30px 上边框向外移动10px，左右边框向外移动20px，下边框向外移动30px
 - 10px 20px 30px 40px 上边框向外移动10px，右边框向外移动20px，下边框向外移动30px，左边框向外移动40px

- 简写方式: `border-image: <border-image-source> || <border-image-slice> [/<border-image-width> || <border-image-repeat>]`

圆角

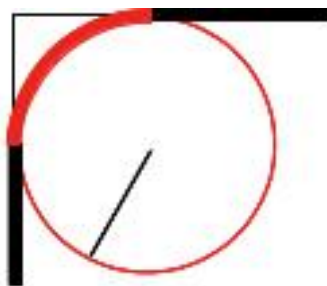
boder-radius

- border-radius: none | <length> {1, 4} [/ <length> {1, 4}] ?
 1. 如果有反斜杠，前面设置的元素圆角水平方向半径，后面是垂直方向半径。
 2. 如果没有反斜杠，表示圆角水平、垂直方向半径相同。



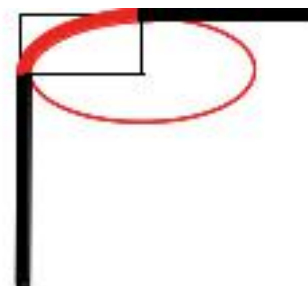
无圆角

border-radius: none



圆形圆角

border-radius: 50px



椭圆形圆角

border-radius: 50px 15px

- border-radius:
 - `<length>{1}` , 四个角的值相同, 也就是4个圆角。
- border-radius:
 - `<length>{2}` , top-left等于bottom-right, 并且取第一个值。top-right等于bottom-left, 取第二个值。
- border-radius:
 - `<length>{3}` , top-left取第一个值, top-right和bottom-left取第二个值, bottom-right取第三个值。
- border-radius:
 - `<length>{4}` , 四个角取不同的值, top-left取第一个, top-right取第二个, bottom-right取第三个, bottom-left取第四个。

1. 如果不想给元素设置圆角，只需要 `border-radius: 0` 即可。
2. `border-radius: 50%`，如果元素是正方形，显示圆形
3. `border-radius: 50%`，如果元素是长方形，显示椭圆

- 属性box-shadow property设置盒子的阴影
- box-shadow:
 - 参数1：投影方式。默认外阴影，inset是内阴影。
 - 参数2：X轴偏移量，可以是负值
 - 参数3：Y轴偏移量，可以是负值
 - 参数4：阴影模糊半径
 - 参数5：阴影扩展半径，可以是负值
 - 参数6：阴影颜色;

五、CSS3 盒子阴影属性

- none: 默认值，元素不设置阴影。
- inset: 阴影类型，可选。如果不设置投影方式外部阴影，如果设置inset就是内 阴影
- x-offset:阴影水平偏移。正值，阴影在元素右侧，负值在元素左侧。
- y-offset:阴影垂直偏移。正式，阴影在元素底部，负值在元素顶部。
- blur-radius: 阴影的模糊半径，可选。只能为正值，如果设置为0，阴影不具备模糊效果。
- spread-radius: 阴影扩展半径。可选参数。其值可正可负。如果取正值，整个阴影都延展扩大，反之取负值，则整个阴影都缩小。
- color: 阴影的颜色。 可选参数

- 单边实影：只需要调整 x-offset 或者 y-offset 即可。
 - box-shadow: 0 -2px #666 上阴影
 - box-shadow: 2px 0 #666 右阴影
 - box-shadow: 0 2px #666 下阴影
 - box-shadow: -2px 0 #666 左阴影
- 单边阴影：调整 x-offset 或者 y-offset 设置偏移，设置模糊半径实现模糊效果。
 - box-shadow: 0 -2px 5px #666 上模糊阴影
 - box-shadow: 2px 0 5px #666 右模糊阴影
 - box-shadow: 0 2px 5px #666 下模糊阴影
 - box-shadow: -2px 0 5px #666 左模糊阴影

- 单边阴影： 设置阴影深度。
 - `box-shadow: 0 -2px 5px 6px #666` 上模糊阴影
 - `box-shadow: 2px 0 5px 6px #666` 右模糊阴影
 - `box-shadow: 0 2px 5px 6px #666` 下模糊阴影
 - `box-shadow: -2px 0 5px 6px #666` 左模糊阴影
 - 阴影扩展半径是正值，与设置相同大小边框（border）的效果相同

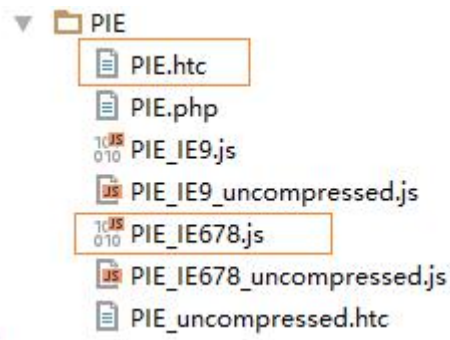
- outline: 元素轮廓
 - 语法 [<'outline-color'> || <'outline-style'> || <'outline-width'>]
 - outline-color 外轮廓颜色
 - outline-style 外轮廓样式
 - outline-width 外轮廓宽度
- outline-offset: 外轮廓偏移量, 不会导致盒子大小改变。
 - 正值: 外偏移 outline-offset: 10px
 - 负值: 内偏移 outline-offset: -10px
- 知识点:
 - 外轮廓不会改变元素盒子大小

- `resize`:
 - `none`: 用户不能调整元素宽高。
 - `both`: 用户可以调整元素宽高。
 - `horizontal`: 用户只能调整元素宽度
 - `vertical`: 用户只能调整元素高度
 - `inherit`: 默认继承
 - 注意事项: `resize` 必须搭配 `overflow: auto;`

解决IE低版本兼容性问题

- border-radius: 圆角
- box-shadow: 阴影
- border-image: 图片边框

- 第一步：下载PIE.js项目压缩文档
- 第二步：解压缩文件。把PIE



- PIE
 - PIE.htc
 - PIE.php
 - PIE_IE9.js
 - PIE_IE9_uncompressed.js
 - PIE_IE678.js
 - PIE_IE678_uncompressed.js
 - PIE_uncompressed.htc

```
<!--[if lte IE 8]>
<script type="text/javascript" src="../PIE/PIE_IE678.js"></script>
<![endif]-->

<!--[if IE 9]>
<script type="text/javascript" src="../PIE/PIE_IE9.js"></script>
<![endif]-->
```

```
border-radius: 0 100px 100px 0;
```

```
behavior: url(../PIE/PIE.htc);
```

- 第三步：编写CSS3样式
- 第四步：behavior: url(path/to/pie_files/PIE.htc);
- 第五步：在IE下查看



模式

文档模式

7

[通过 F12 开发者工具栏](#)

浏览器配置文件

桌面

用户代理字符串

Internet Explorer 7

