

HTML5 文本属性 圆角

四、圆角与文本属性

圆角 border-radius

文本属性

white-space : 空格、缩进、换行的处理方式

2. Word-wrap:break-word 与 word-break:break-all

文本书写模式

文本超出 : text-overflow

文字阴影 : text-shadow

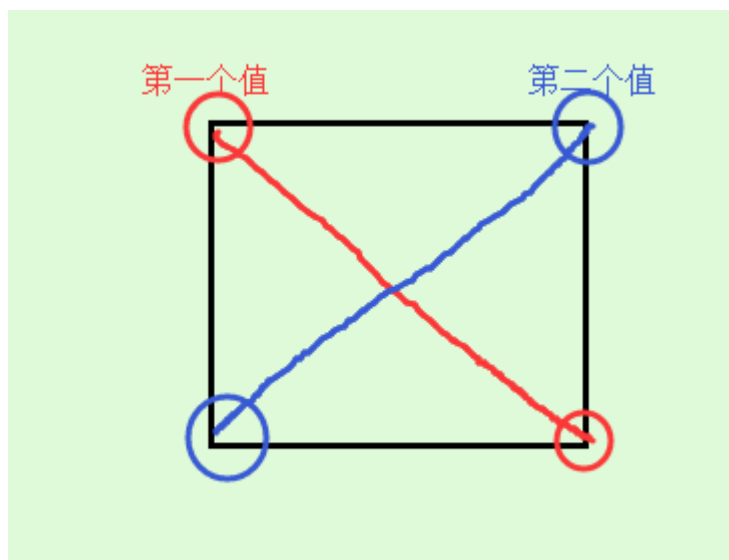
文字描边、文字填充

文字大小写 text-transform

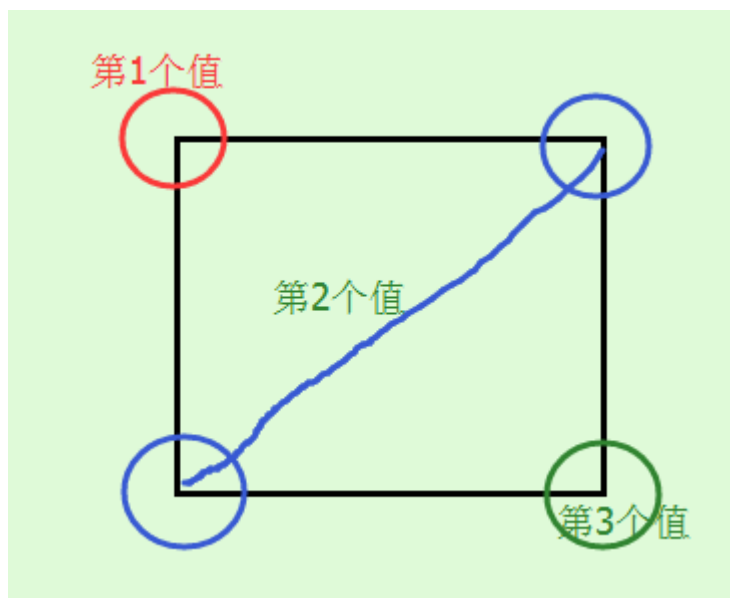
四、圆角与文本属性

圆角 border-radius

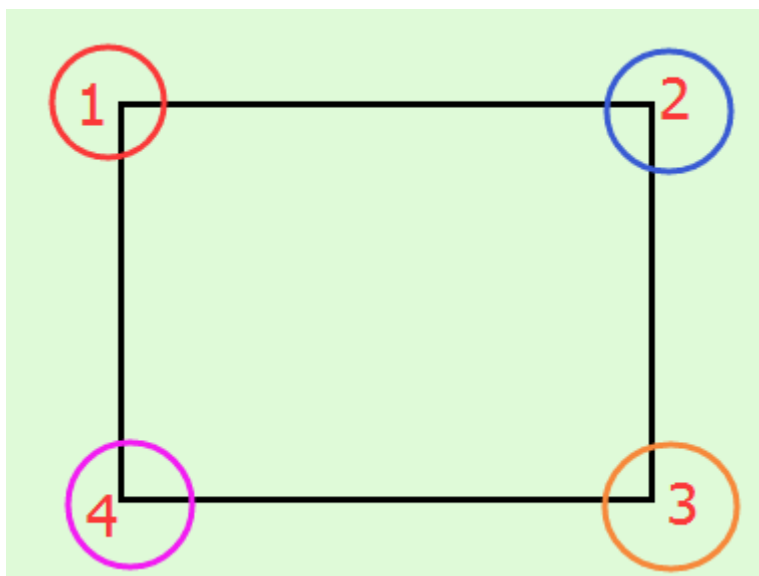
- 如果是一个值，那么 top-left、top-right、bottom-right、bottom-left 四个值相等。
- 如果是两个值，那么 top-left和bottom-right相等，为第一个值，top-right和bottom-left值相等，为第二个值。



- 如果是三个值，那么第一个值是设置top-left，而第二个值是 top-right 和 bottom-left 并且他们会相等,第三个值是设置 bottom-right。



- 如果是四个值，那么第一个值是设置 top-left，而第二个值是 top-right 第三个值 bottom-right 第四个值是设置 bottom-left



`border-radius`可以单独指定水平和垂直半径，只需要用一个斜杠(水平/垂直)分割即可。

除了上述的简写外，还可以和border一样，分别写四个角，如下：

`border-top-left-radius`: //左上角

`border-top-right-radius`: //右上角

`border-bottom-right-radius`: //右下角

`border-bottom-left-radius`: //左下角

分别是水平方向和竖直方向半径，第二值省略的情况下，水平方向和竖直方向的半径相等

文本属性

white-space：空格、缩进、换行的处理方式

1. `normal` 默认 忽略多个空格/缩进/换行只留一个
2. `nowrap` 控制文本不换行
3. `pre` 空白/缩进/换行 会被浏览器保留
4. `pre-line` 合并空白/缩进 保留换行符
5. `pre-wrap` 保留空白/缩进 正常换行

2. Word-wrap:break-word 与 word-break:break-all

两者究竟有什么样的区别呢。我们看看官方的解释：

word-wrap

The `word-wrap` CSS property is used to specify whether or not the browser is allowed to break lines within words in order to prevent overflow when an otherwise unbreakable string is too long to fit.

word-break

The `word-break` CSS property is used to specify how (or if) to break lines within words.

看看翻译后的：

`word-wrap:`

css的 `word-wrap` 属性用来标明是否允许浏览器在单词内进行断句，这是为了防止当一个字符串太长而找不到它的自然断句点时产生溢出现象。 从上一个单词换行开始

`word-break:`

css的 `word-break` 属性用来标明怎么样进行单词内的断句。 可以从上一个的结尾开始

两者对中文无效

文本书写模式

`direction`：规定文本的方向

`unicode-bidi`：设置文本的方向

1. `direction: rtl => right to left`
2. `direction: ltr => left to right`
3. `unicode-bidi: bidi-override`

`writing-mode` :设置或检索对象的内容块固有的书写方向

1. `horizontal-tb`: 水平方向自上而下的书写方式。即 `left-right-top-bottom` (类似IE私有值`lr-tb`)
2. `vertical-rl`: 垂直方向自右而左的书写方式。即 `top-bottom-right-left` (类似IE私有值`tb-rl`)
3. `vertical-lr`: 垂直方向自左而右的书写方式。即 `top-bottom-left-right`
4. `lr-tb`: 左-右, 上-下。对象中的内容在水平方向上从左向右流入, 后一行在前一行的下面。 所有的字形都是竖直向上的。这种布局是罗马语系使用的 (IE)
5. `tb-rl`: 上-下, 右-左。对象中的内容在垂直方向上从上向下流入, 自右向左。后一竖行在前一竖行的左面。全角字符是竖直向上的, 半角字符如拉丁字母或片假名顺时针旋转90度。这种布局是东亚语系通常使用的 (IE)

文本超出 : text-overflow

主要用于文本超出后显示省略号, 结合`white-space`与`overflow`使用

1. `white-space: nowrap`; /*文本不换行*/
2. `overflow: hidden`; /*超出隐藏*/
3. `text-overflow: ellipsis`; /*显示省略号*/

上面的是单行的显示, 多行的显示省略号怎么弄呢? 结合以下的几个属性

1. `display: -webkit-box`;
2. `-webkit-box-orient: vertical`;
3. `-webkit-line-clamp: 5`; /*设置文本显示的行数*/

当然, 你也可以使用js实现

文字阴影 : text-shadow

参数: x轴偏移量 y轴偏移量 模糊度 颜色

例子:

浮雕例子:

1. `color: #fff; text-shadow: 2px 2px 4px #000`;

立体字:

1. `color:#fff;`
2. `text-shadow:0 1px 0 #ccc,0 2px 0 #9c9c9c,0 3px 0 #bbb,0 4px 0 #9b9b9b,0 5px 0 #aaa;`

模糊字：

1. `color:rgba(0,0,0,0); text-shadow:0 0 20px blue;`

霓虹灯：

1. `text-shadow:0px 0px 10px #f3c,0px 0px 20px #f39,0px 0px 40px #f33,0px 0px 60px #f30;`

文字描边、文字填充

text-stroke：宽度 颜色；描边，加上兼容性前缀

text-stroke-width：宽度

text-stroke-color：颜色

text-fill-color: 颜色

文字大小写 text-transform

四个值：

1. `none`：默认值
2. `capitalize`：将每个单词的第一个字母转换为大写
3. `uppercase`：转换为大写
4. `lowercase`：转换为小写