

# CSS属性-盒子模型

@M了个J

<https://github.com/CoderMJLee>

<https://space.bilibili.com/325538782>



实力IT教育 www.520it.com



- HTML中的每一个元素都可以看做是一个盒子



# 盒子模型 (Box Model)

■ HTML中的每一个元素都可以看做是一个盒子，如右下图所示，可以具备这4个属性

□ 内容 (content)

✓ 盒子里面装的东西

□ 内边距 (padding)

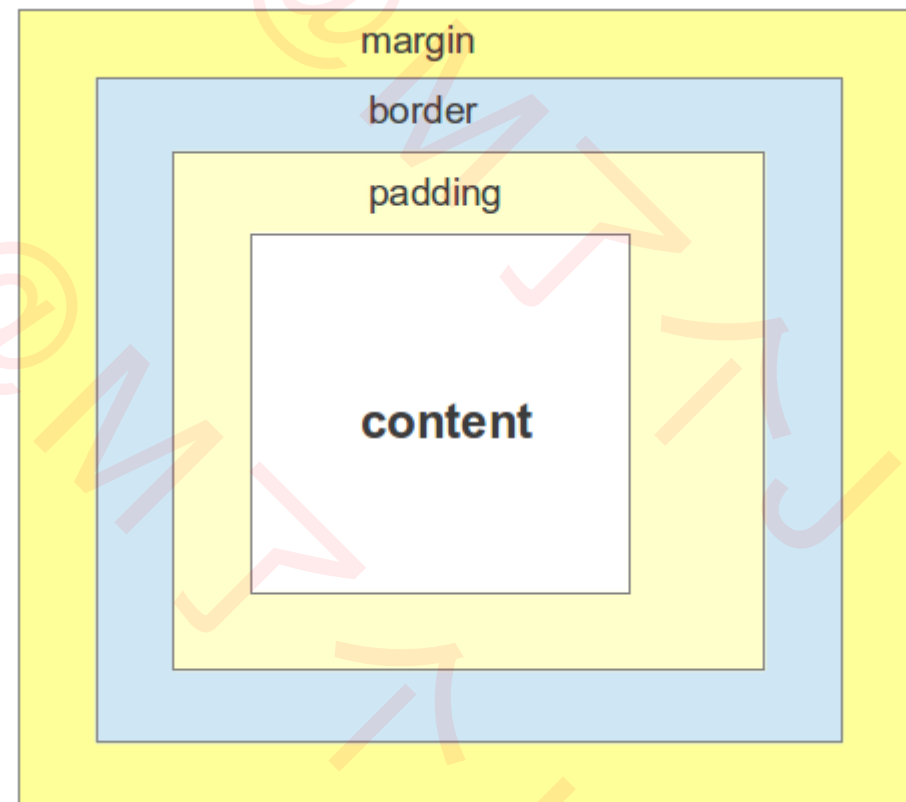
✓ 怕盒子里装的东西损坏而添加的泡沫或者其它抗震的辅料

□ 边框 (border)

✓ 就是盒子的边框，比如木盒子四周的木板

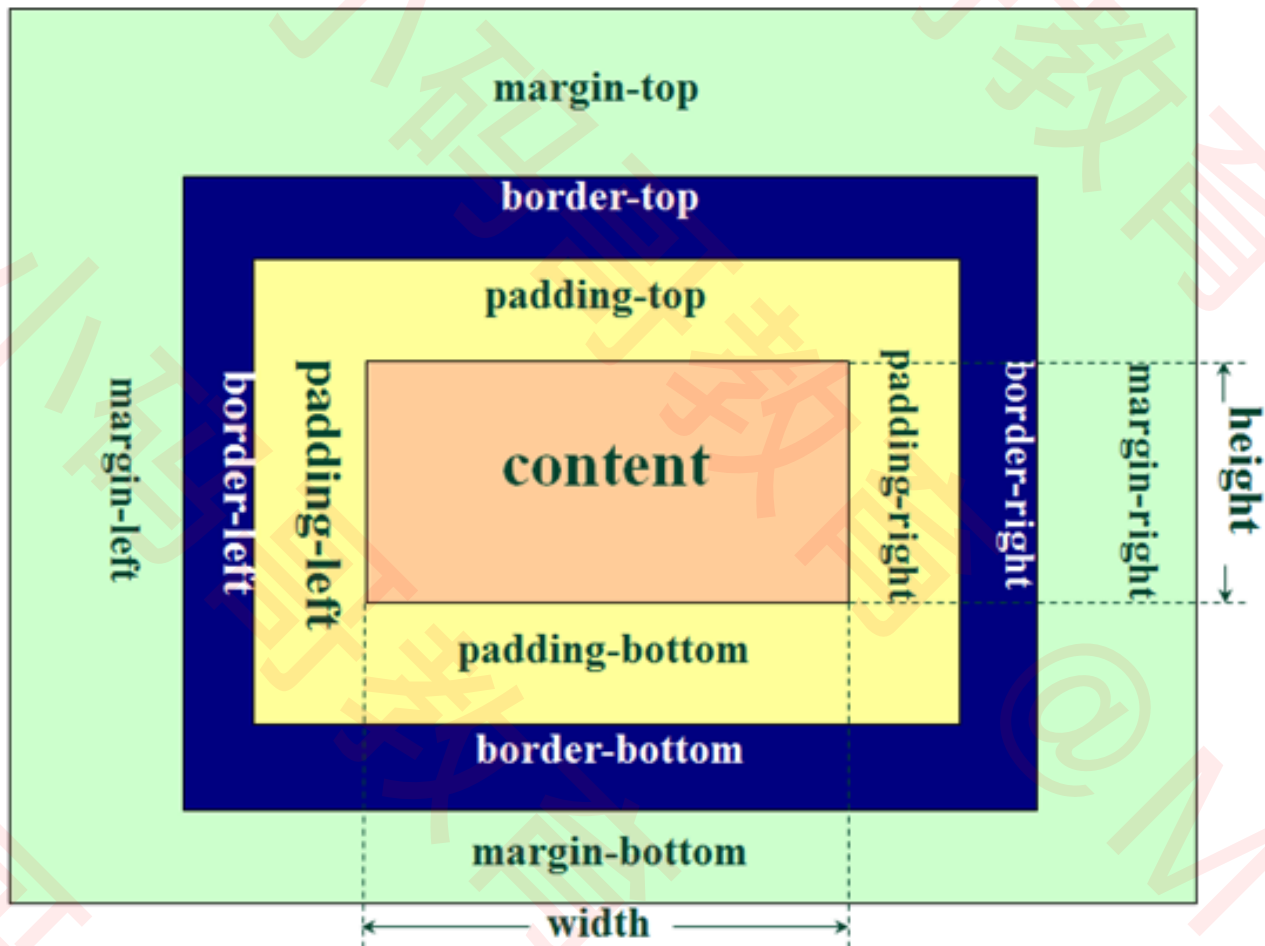
□ 外边距 (margin)

✓ 为了方便取出，盒子之间保留一定的空隙

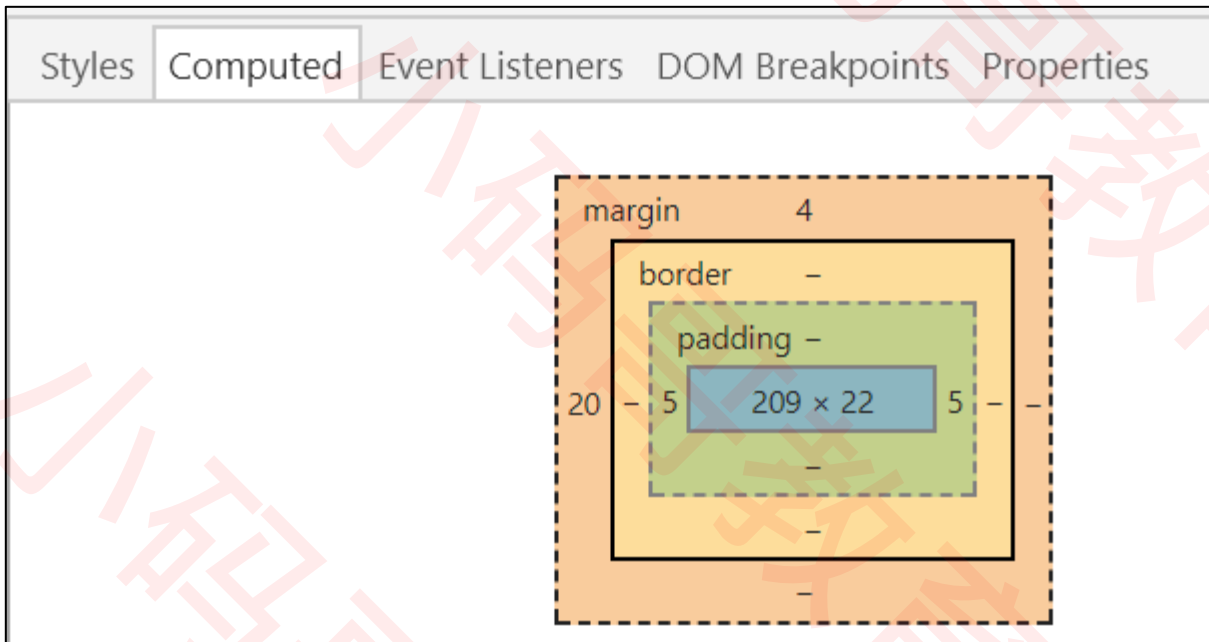


# 盒子模型

- 默认盒子模型如下图所示



# 浏览器开发者工具中看到的盒子模型



一个元素的实际占用宽度 = border-left + padding-left + width + padding-right + border-right

一个元素的实际占用高度 = border-top + padding-top + height + padding-bottom + border-bottom

一个元素的空间宽度 = margin-left + border-left + padding-left + width + padding-right + border-right + margin-right

一个元素的空间高度 = margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom

# 宽度、高度相关属性

- width: 宽度
- min-width: 最小宽度, 无论内容多少, 宽度都大于或等于min-width
- max-width: 最大宽度, 无论内容多少, 宽度都小于或等于max-width
- height: 高度
- min-height: 最小高度, 无论内容多少, 高度都大于或等于min-height
- max-height: 最大高度, 无论内容多少, 高度都小于或等于max-height

# 内边距相关属性

- padding-left: 左内边距
- padding-right: 右内边距
- padding-top: 上内边距
- padding-bottom: 下内边距
- padding: 是padding-top、padding-right、padding-bottom、padding-left的简写属性

# padding

```
div { padding: 10px 20px 30px 40px; }
```

```
div {  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 40px;  
}
```

```
div { padding: 10px 20px; }
```

```
div {  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 10px;  
  padding-left: 20px;  
}
```

```
div { padding: 10px 20px 30px; }
```

```
div {  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 20px;  
}
```

```
div { padding: 10px; }
```

```
div {  
  padding-top: 10px;  
  padding-right: 10px;  
  padding-bottom: 10px;  
  padding-left: 10px;  
}
```



# padding的取值规律

- 按照顺时针方向设值：top、right、bottom、left
- 如果缺少left, 那么left就使用right的值
- 如果缺少bottom, 那么bottom就使用top的

# 外边距相关属性

- margin-left: 左外边距
- margin-right: 右外边距
- margin-top: 上外边距
- margin-bottom: 下外边距
- margin: 是margin-top、margin-right、margin-bottom、margin-left的简写属性

# margin

```
div { margin: 10px 20px 30px 40px; }
```

```
div {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
}
```

```
div { margin: 10px 20px; }
```

```
div {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 10px;  
  margin-left: 20px;  
}
```

```
div { margin: 10px 20px 30px; }
```

```
div {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 20px;  
}
```

```
div { margin: 10px; }
```

```
div {  
  margin-top: 10px;  
  margin-right: 10px;  
  margin-bottom: 10px;  
  margin-left: 10px;  
}
```

# 上下margin传递

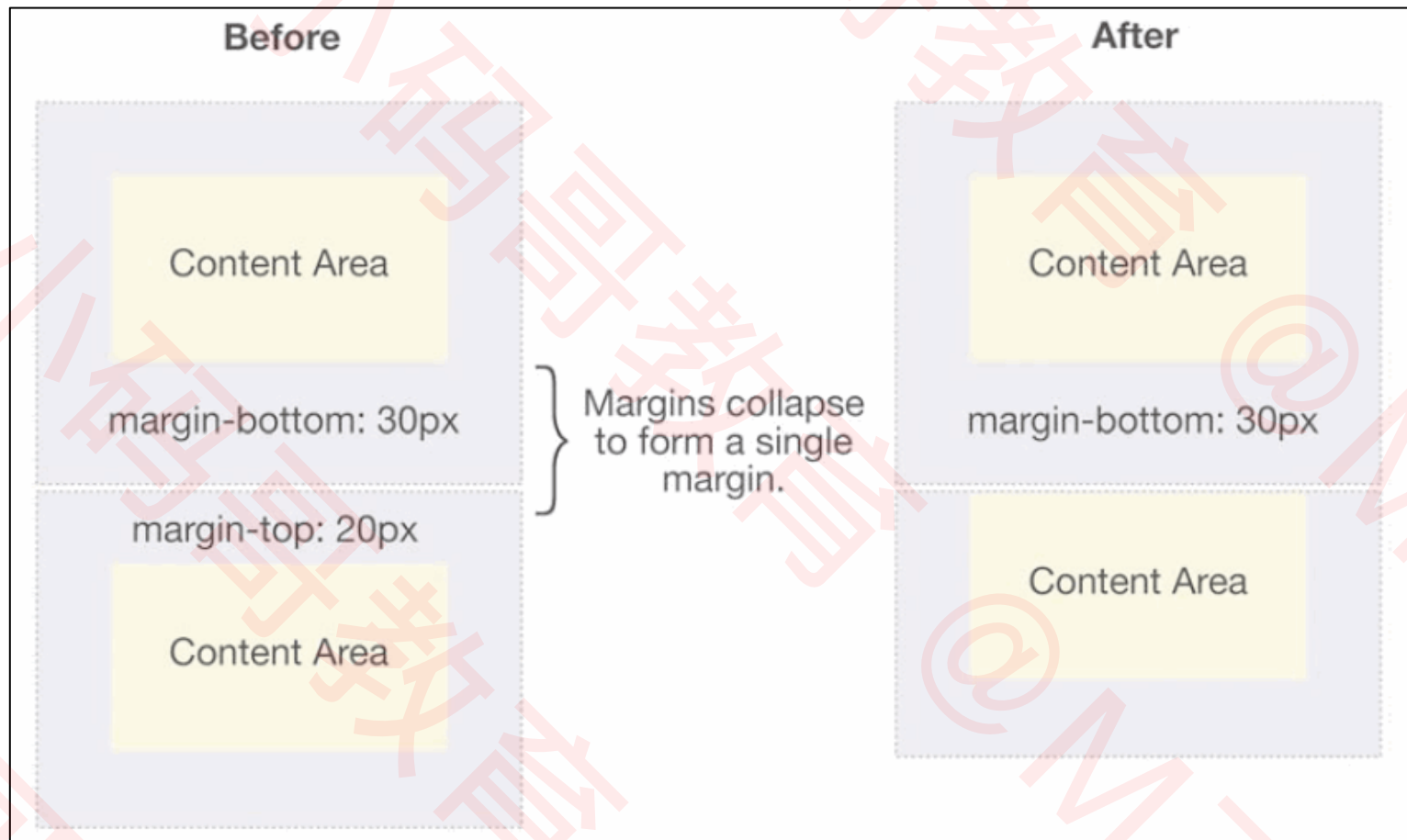
- margin-top传递
  - 如果块级元素的顶部线和块级父元素的顶部线重叠，那么这个块级元素的margin-top值会传递给父元素
- margin-bottom传递
  - 如果块级元素的底部线和块级父元素的底部线重叠，并且父元素的高度是auto，那么这个块级元素的margin-bottom值会传递给父元素
- 水平方向上的margin是永远不会发生传递现象
- 如何防止出现传递问题？
  - 给父元素设置padding-top\padding-bottom
  - 给父元素设置border
  - 给父元素或者子元素设置display: inline-block
  - 以后理解得更深入，还会学到其他解决方法
- 建议
  - margin一般是用来设置兄弟元素之间的间距
  - padding一般是用来设置父子元素之间的间距

# 上下margin折叠

- 垂直方向上相邻的2个margin (margin-top、margin-bottom) 有可能会合并为1个margin, 这种现象叫做collapse (折叠)
- 水平方向上的margin (margin-left、margin-right) 永远不会collapse
- 折叠后最终值的计算规则
  - 如果都是正数, 最终值是: 绝对值最大的那个正数值
  - 如果都是负数, 最终值是: 绝对值最大的那个负数值
  - 如果正数、负数都有, 最终值是: 最大正数和最小负数相加
- 如何防止margin collapse?
  - 只设置其中一个元素的margin
  - 条件允许的话, 使用padding取代margin
  - 以后理解得更深入, 还会学到其他解决方法

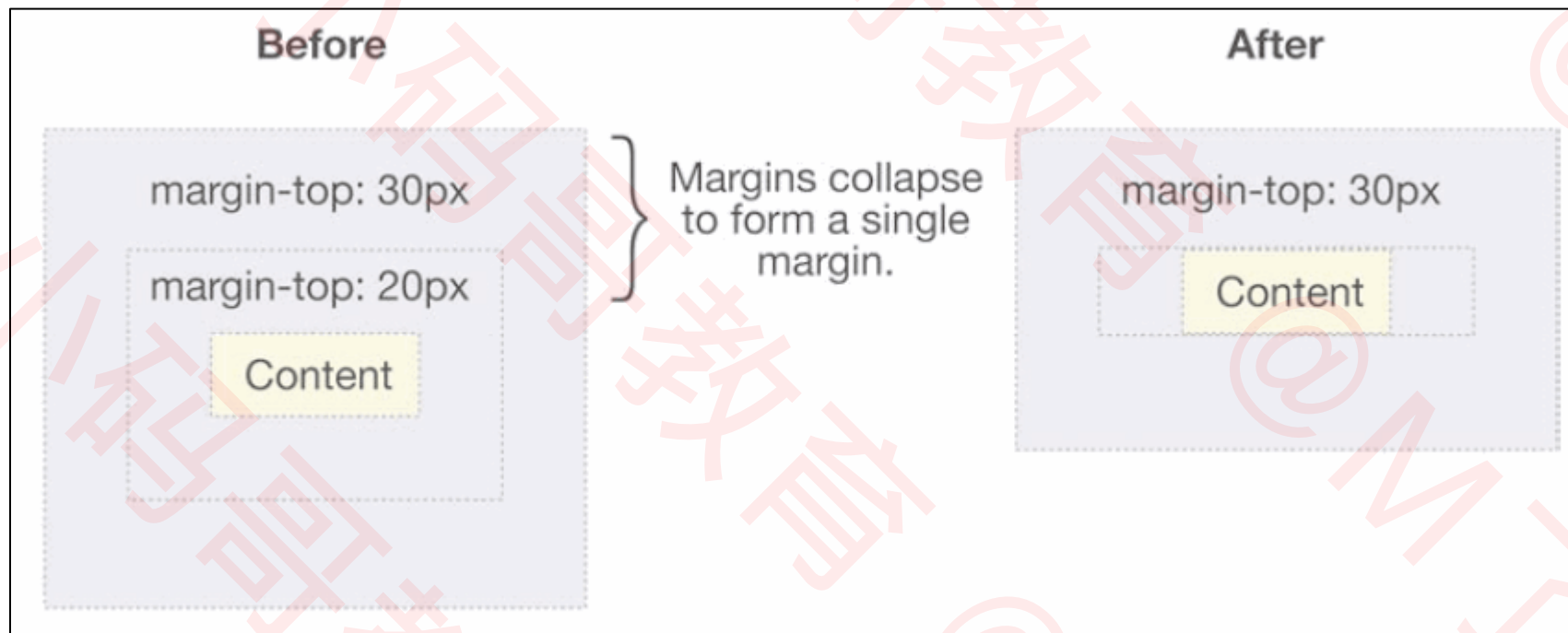
# 上下margin折叠

- 两个兄弟块级元素之间上下margin的折叠



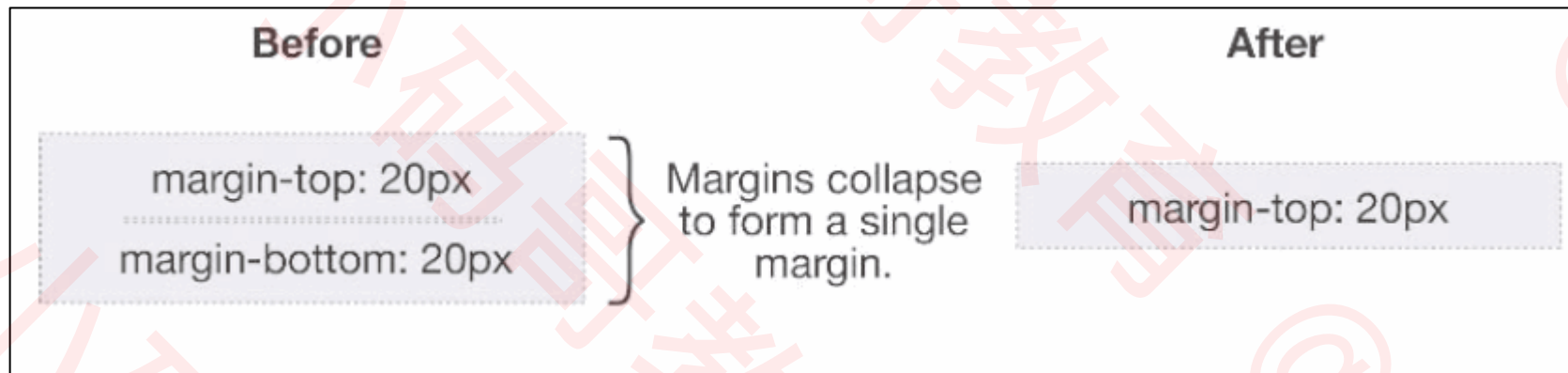
# 上下margin折叠

## ■ 父子块级元素之间margin的折叠



# 上下margin折叠

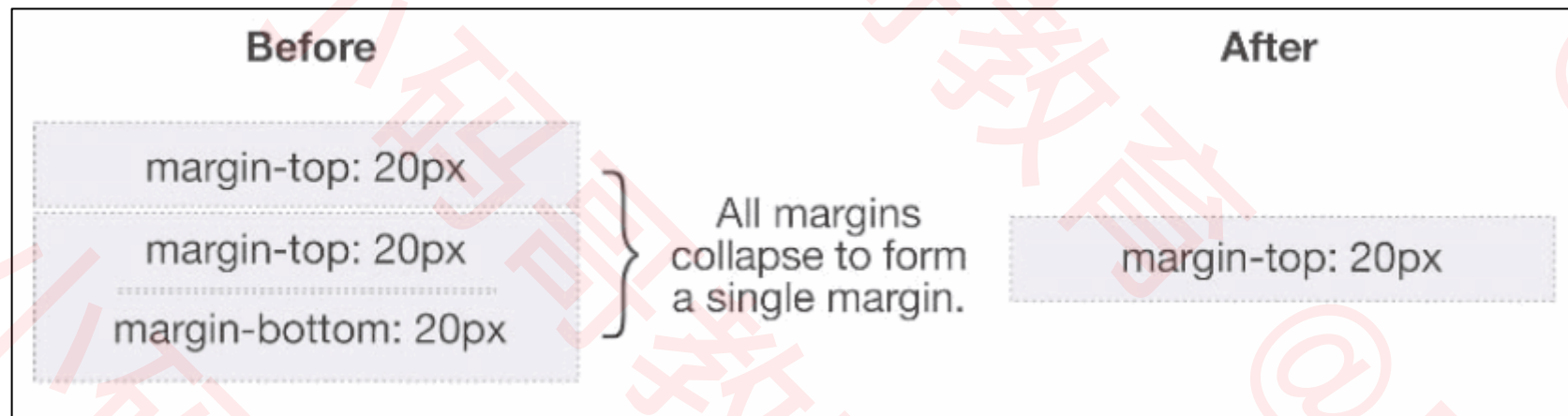
## ■ 无内容块级元素内部margin的折叠





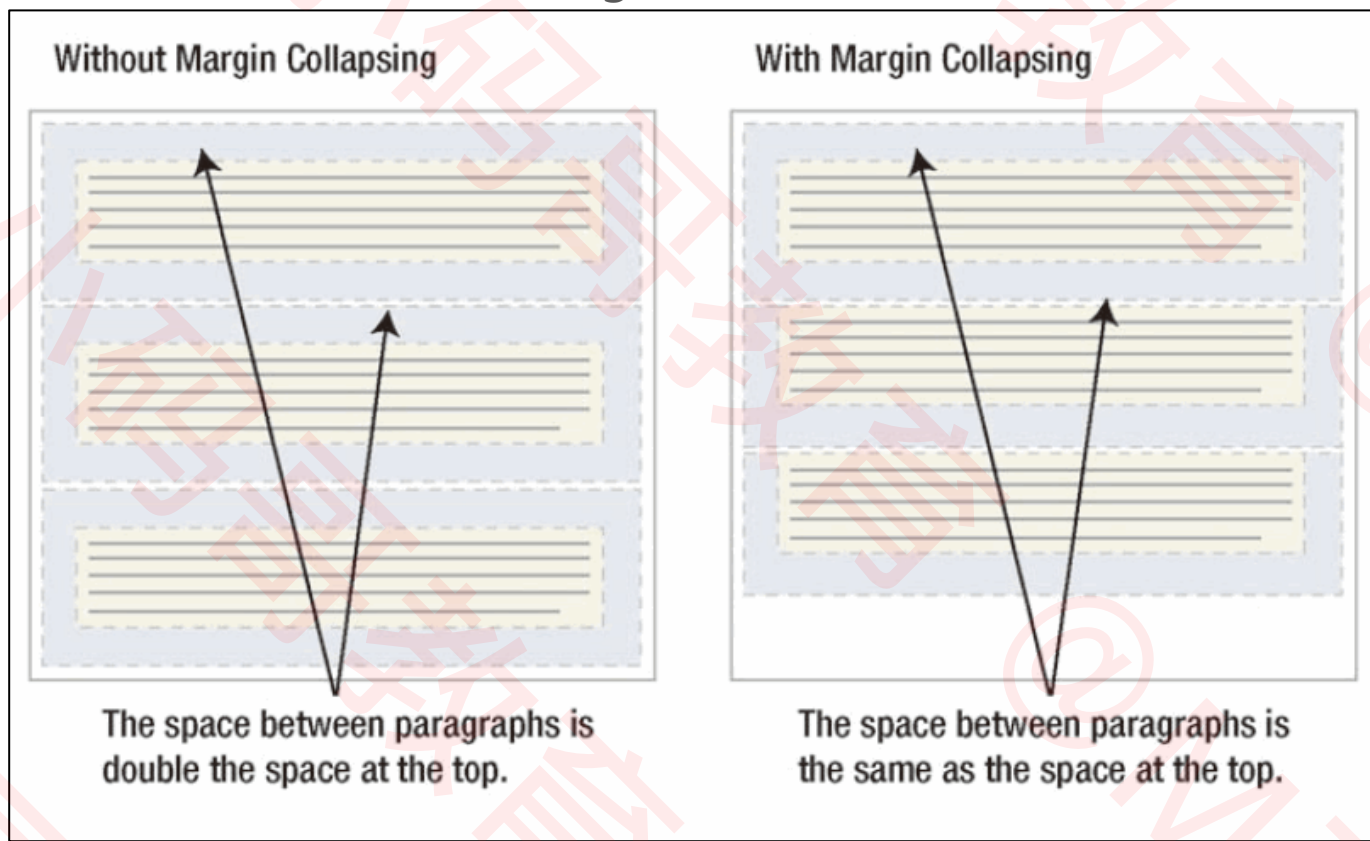
# 上下margin折叠

- 无内容块级元素之间margin的折叠（折叠是可以连续的）



# 上下margin折叠

- 块级元素折叠问题看似有点莫名其妙，实际上还是有实用之处的
- 比如连续段落之间的margin，恰好需要这种折叠效果



# border (边框)



选择颜色



珍珠黑



流光银



玫瑰金



J33 白色



增值保障



意外修2年 ¥29



1年内换新 ¥59



全保2年 ¥49



白条分期

不分期

¥83×3期

¥41.5×6期

¥21.37×12期

¥11.62×24期



# 边框相关的属性

## ■ 边框宽度

- border-top-width、border-right-width、border-bottom-width、border-left-width
- border-width是上面4个属性的简写属性

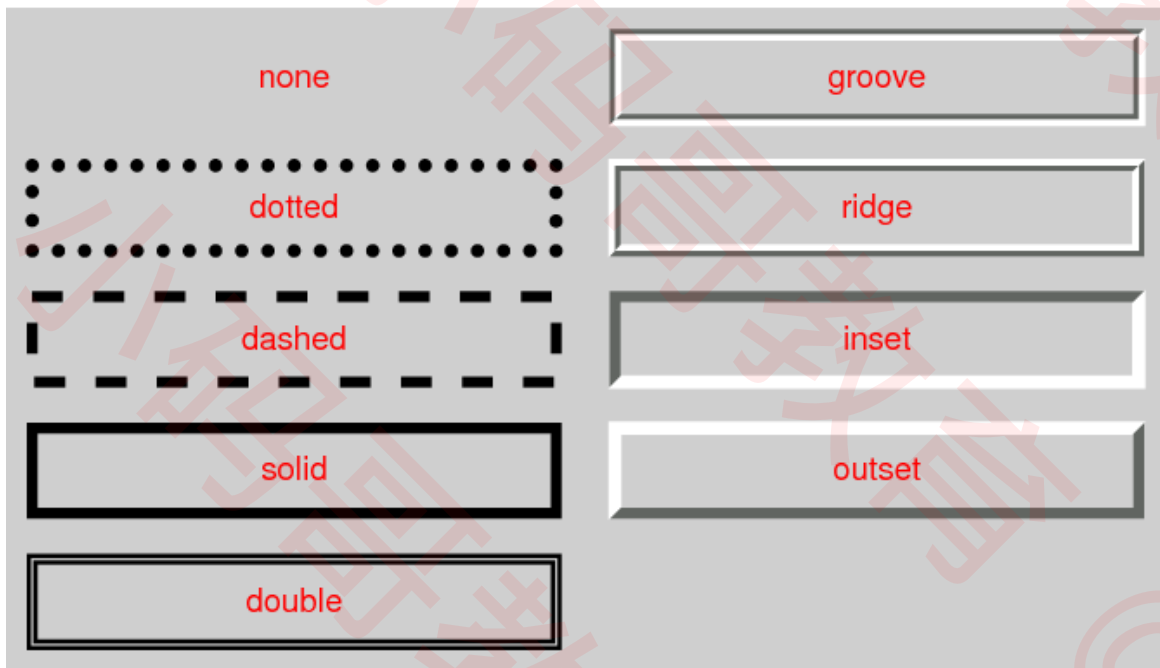
## ■ 边框颜色

- border-top-color、border-right-color、border-bottom-color、border-left-color
- border-color是上面4个属性的简写属性

## ■ 边框样式

- border-top-style、border-right-style、border-bottom-style、border-left-style
- border-style是上面4个属性的简写属性

# 边框样式的取值

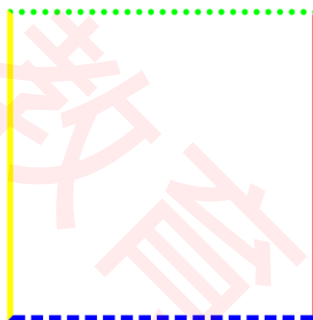


- **none**: 没有边框，边框颜色、边框宽度会被忽略
- **hidden**: 与“none”类似，多用在表格上，用于解决边框冲突
- **dotted**: 边框是一系列的点
- **dashed**: 边框是一条虚线
- **solid**: 边框是一条实线
- **double**: 边框有两条实线。两条线宽和其中的空白的宽度之和等于border-width的值
- **groove**: 边框看上去好像是雕刻在画布之内
- **ridge**: 和groove相反，边框看上去好像是从画布中凸出来
- **inset**: 该边框使整个框看上去好像是嵌在画布中
- **outset**: 和inset相反，该边框使整个框看上去好像是从画布中凸出来

# 边框相关的属性

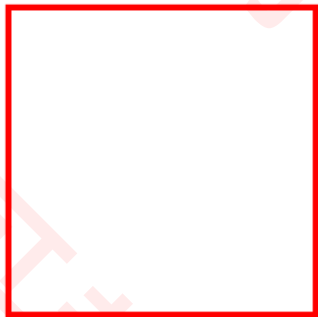
- border-top、border-right、border-bottom、border-left

```
border-top: 2px #0f0 dotted ;  
border-right: 2px solid #f00;  
border-bottom: dashed 2px #00f;  
border-left: #ff0 solid 2px;
```



- 边框颜色、宽度、样式的编写顺序任意
- border: 统一设置4个方向的边框

```
border: 2px solid #f00;
```



# 边框的形状

- 边框的形状可能是
- 矩形、梯形、三角形等形状

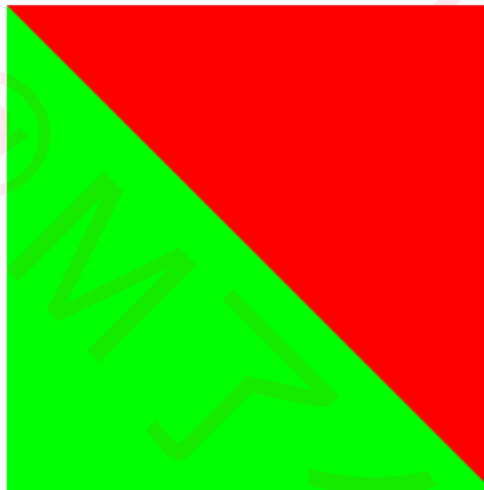
```
border-top: 5px solid #f00;
```



```
border-top: 5px solid #f00;  
border-left: 5px solid #0f0;
```

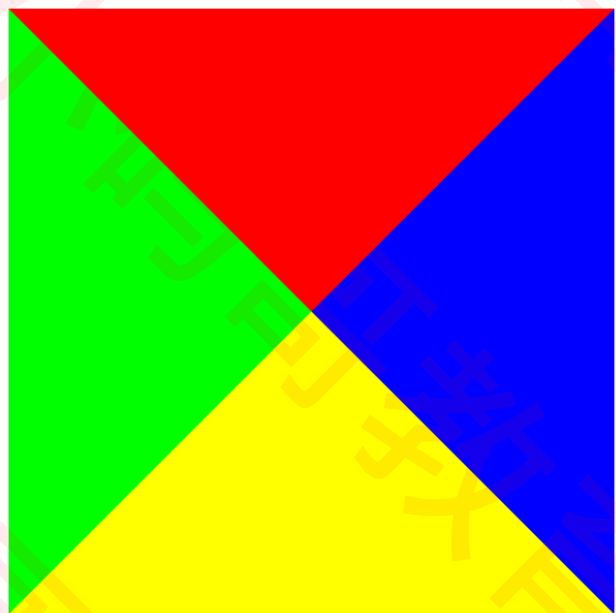


```
border-top: 50px solid #f00;  
border-left: 50px solid #0f0;  
width: 0;  
height: 0;
```

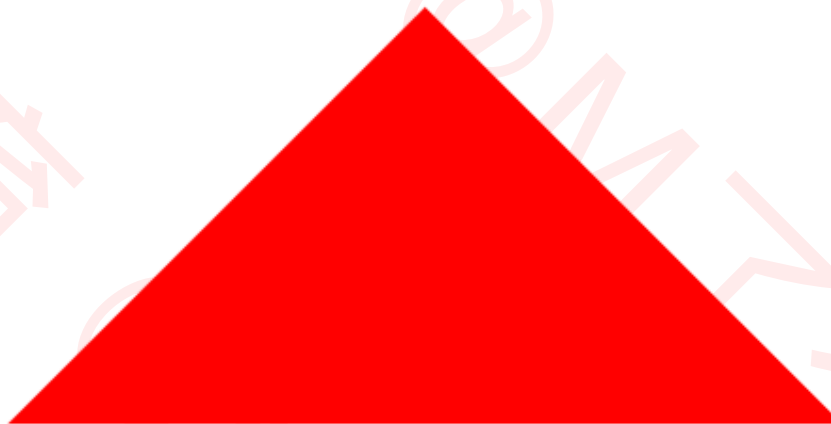


# 边框妙用-实现三角形

```
border-top: 50px solid #f00;  
border-left: 50px solid #0f0;  
border-right: 50px solid #00f;  
border-bottom: 50px solid #ff0;  
width: 0;  
height: 0;
```



```
border-top: 50px solid #f00;  
border-left: 50px solid transparent;  
width: 0;  
height: 0;  
background-color: transparent;  
transform: rotate(-45deg);
```





# 边框妙用-实现双色平分

```
div {  
  background: #DDD;  
  width: 100px;  
  height: 50px;  
  border-bottom: 50px solid #111;  
}
```



# 行内级非替换元素的注意点

- 以下属性对行内级非替换元素不起作用
  - width、height、margin-top、margin-bottom
- 以下属性对行内级非替换元素的效果比较特殊
  - padding-top、padding-bottom、border-top、border-bottom

抽奖赢空调

2件199

厨房电器

下载模板

★ 收藏

注册

登录



参加上传比赛  
赢取现金大奖

去参加比赛

JOBO - THE JOB BOT;)



Jobo - the Job Bot;) to help  
you to find your next job.

# CSS属性 - border-\*-\*-radius

- 圆角半径相关的属性有

*border-top-left-radius, border-top-right-radius,  
border-bottom-right-radius, border-bottom-left-radius*

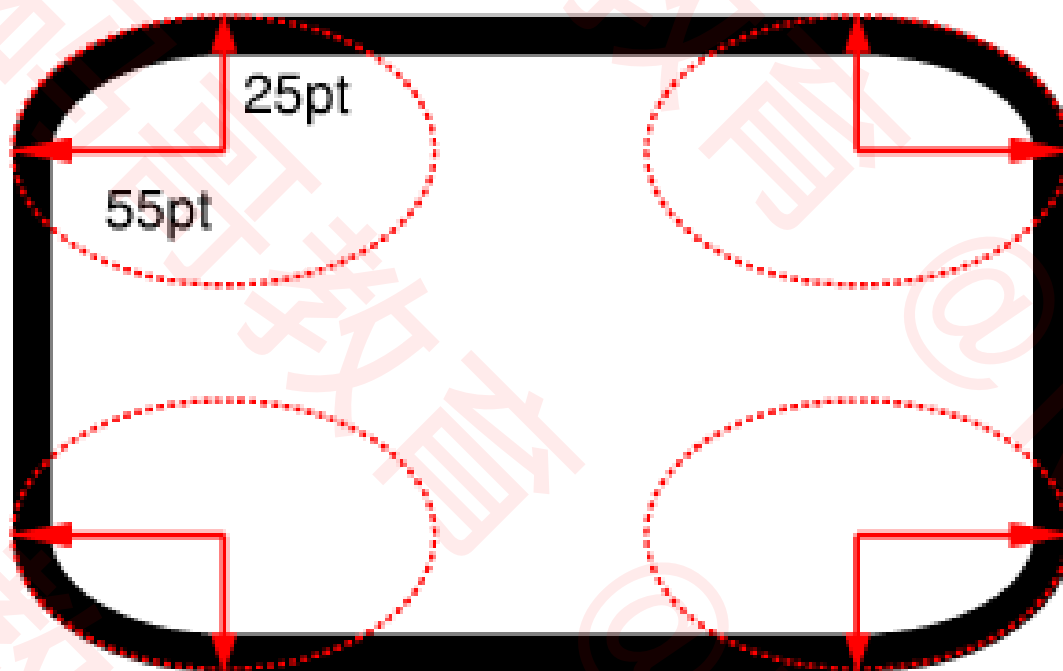
```
border-top-left-radius: 40px 20px;
```

- border-\*-\*-radius定义的是四分之一椭圆的半径
  - 第1个值是水平半径
  - 第2个值是垂直半径（如果不设置，就跟随水平半径的值）

```
border-top-left-radius: 50%;
```

- 还可以设置百分比
  - 参考的是border-box的宽度和高度（也就是元素的实际占用尺寸，border+padding+width\height）

**border-top-left-radius: 55pt 25pt**



# CSS属性 - border-radius

- border-radius是一个缩写属性

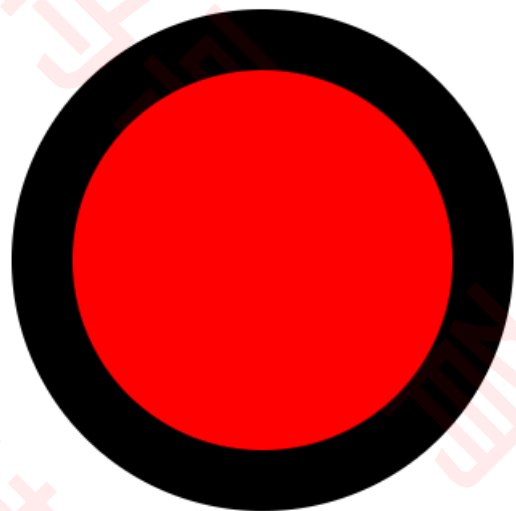
```
border-radius: 10px 20px 30px 40px/15px 25px 35px 45px;
```

- 斜线/前面是水平半径，后面是垂直半径
- 4个值的顺序是top-left、top-right、bottom-right、bottom-left (顺时针方向)
  - 如果bottom-left没设置，就跟随top-right
  - 如果bottom-right没设置，就跟随top-left
  - 如果top-right没设置，就跟随top-left

# CSS属性 - border-radius

```
div {  
  width: 90px;  
  height: 90px;  
  border: 30px solid #000;  
  padding: 50px;  
  background-color: #f00;  
  border-radius: 50%;  
}
```

- border-radius大于或等于50%时，就会变成一个圆



# CSS属性 - outline

- outline表示元素的外轮廓
  - 不占用空间
  - 默认显示在border的外面
  - 每个部位都是完整连接的，不会像border那样有可能会断开（比如行内级非替换元素的换行）
- outline相关属性有
  - outline-width
  - outline-style: 取值跟border的样式一样，比如solid、dotted等
  - outline-color
  - outline: outline-width、outline-style、outline-color的简写属性，跟border用法类似
  - outline-offset: 设置outline和border之间的间距
- 应用实例
  - 去除a元素、input元素的focus轮廓效果



# CSS属性 - box-shadow

- box-shadow属性可以设置一个或者多个阴影
- 每个阴影用<shadow>表示
- 多个阴影之间用逗号,隔开, 从前到后叠加
- <shadow>的常见格式如下

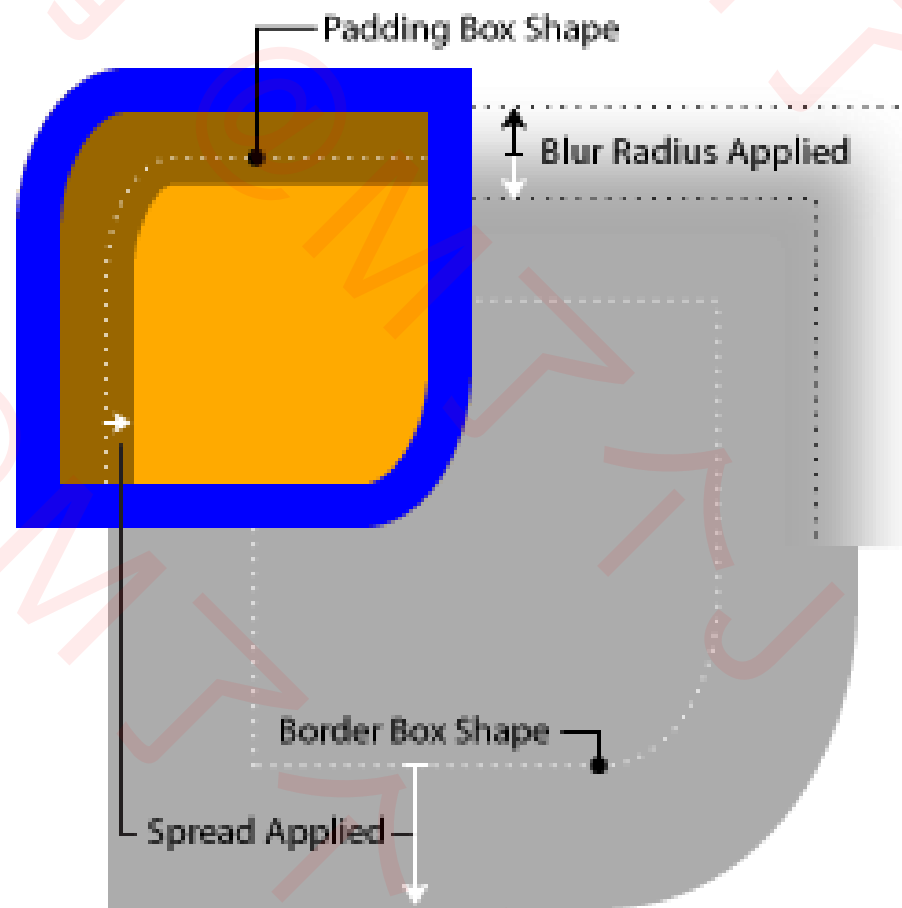
# <shadow>

*<shadow>* = inset? && <length>{2,4} && <color>?

- 第1个<length>：水平方向的偏移，正数往右偏移
- 第2个<length>：垂直方向的偏移，正数往下偏移
- 第3个<length>：模糊半径 (blur radius)
- 第4个<length>：延伸距离
- <color>：阴影的颜色，如果没有设置，就跟随color属性的颜色
- inset：外框阴影变成内框阴影

# box-shadow示例

```
width: 100px; height: 100px;  
border: 12px solid blue; background-color: orange;  
border-top-left-radius: 60px 90px;  
border-bottom-right-radius: 60px 90px;  
box-shadow: 64px 64px 12px 40px rgba(0,0,0,0.4) inset,  
12px 12px 0px 8px rgba(0,0,0,0.4) ;
```



# box-shadow示例

```
border:5px solid blue;  
background-color:orange;  
width: 144px;  
height: 144px;
```

```
box-shadow:  
  rgba(0,0,0,0.4)  
  10px 10px;
```

```
box-shadow:  
  rgba(0,0,0,0.4)  
  10px 10px  
  inset
```

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

```
border-radius: 0;
```

```
box-shadow:  
  rgba(0,0,0,0.4)  
  10px 10px 0  
  10px /* spread */
```

```
box-shadow:  
  rgba(0,0,0,0.4)  
  10px 10px 0  
  10px /* spread */  
  inset
```



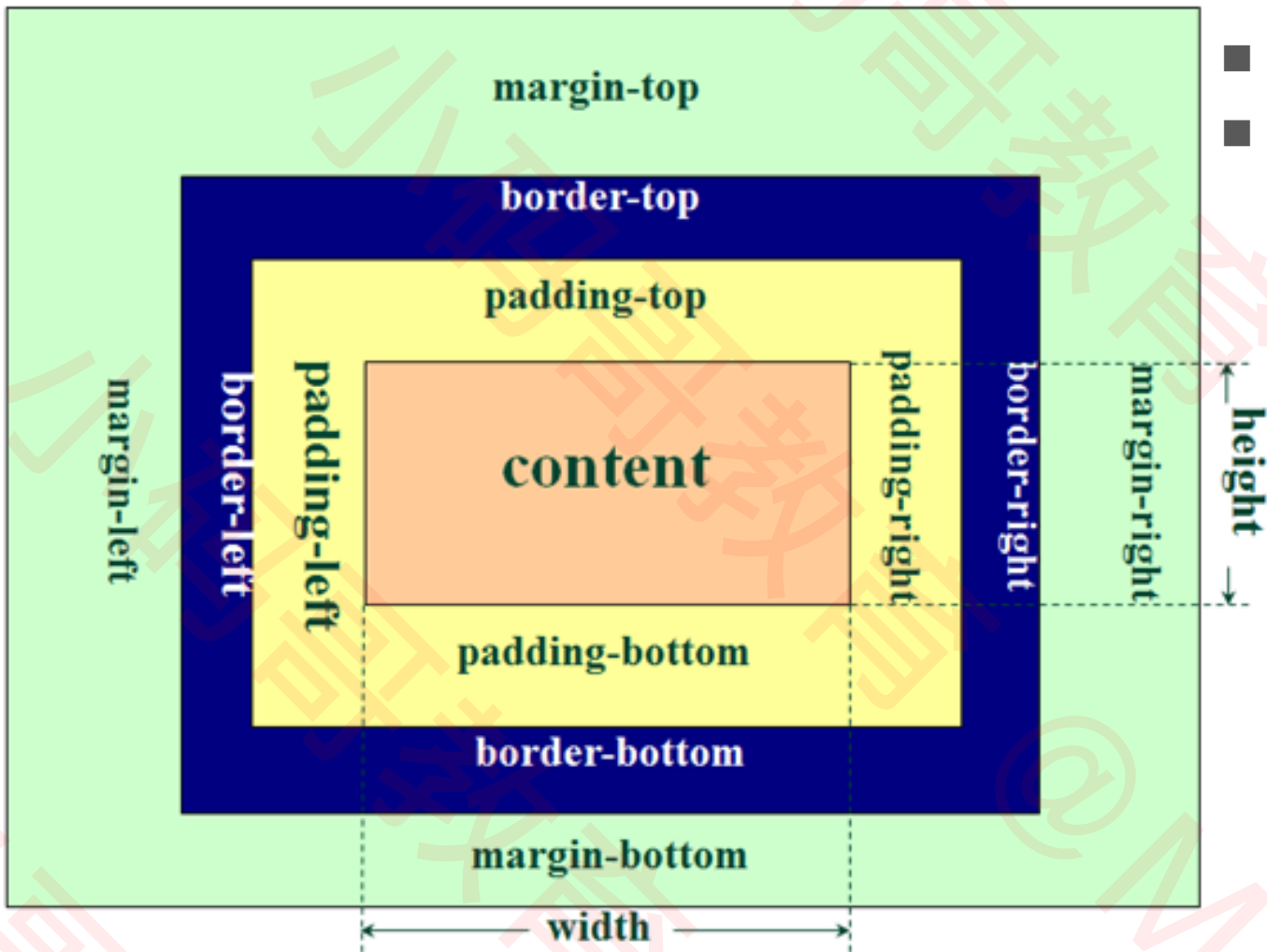
# CSS属性 - text-shadow

- text-shadow用法类似于box-shadow，用于给文字添加阴影效果
- text-shadow同样适用于::first-line、::first-letter

# CSS属性 - box-sizing

- box-sizing用来设置盒子模型中宽高的行为
- content-box
  - ✓ padding、border都布置在width、height外边
- border-box
  - ✓ padding、border都布置在width、height里边

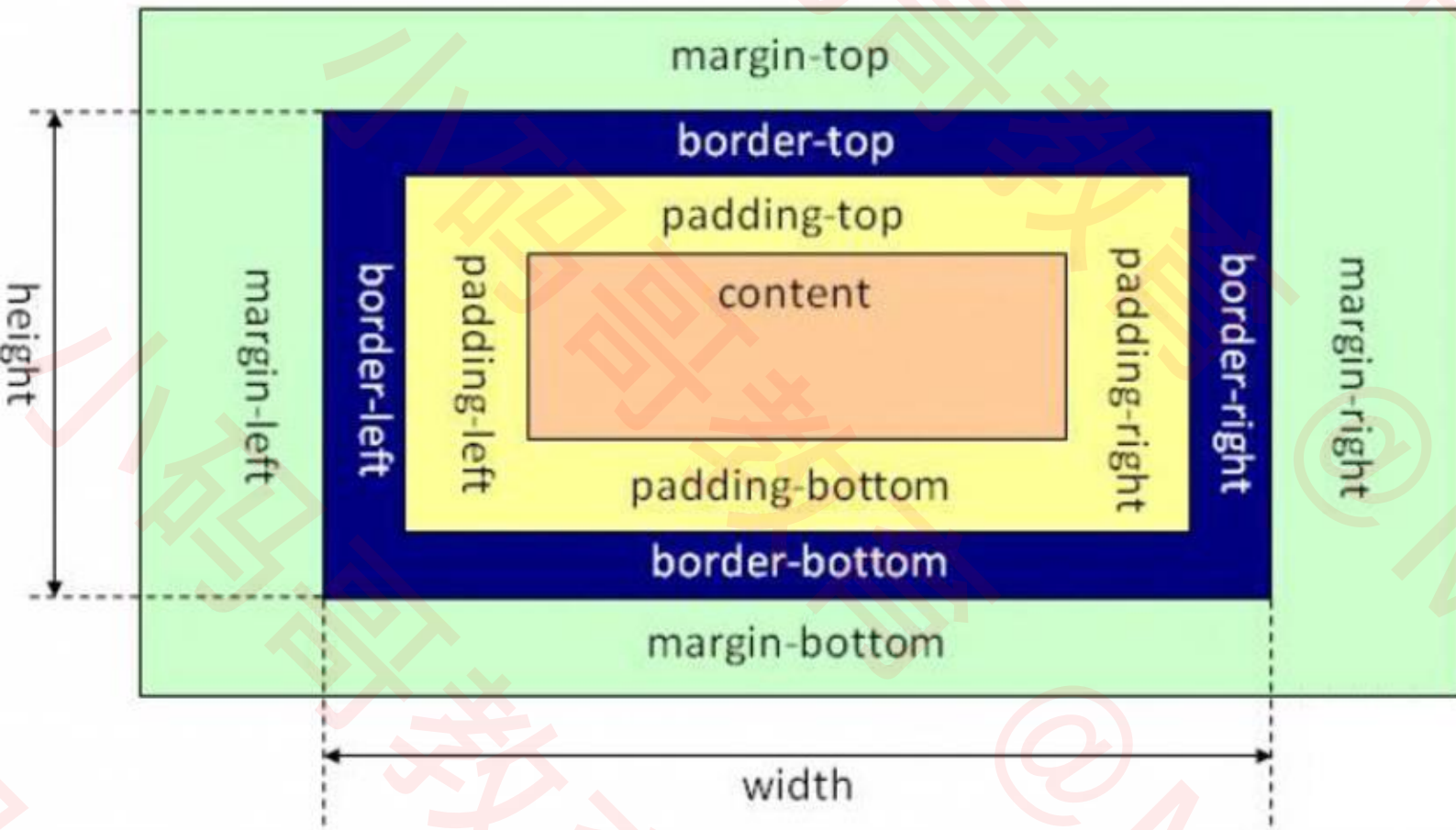
# box-sizing: content-box;



- 元素的实际占用宽度 = border + padding + width
- 元素的实际占用高度 = border + padding + height



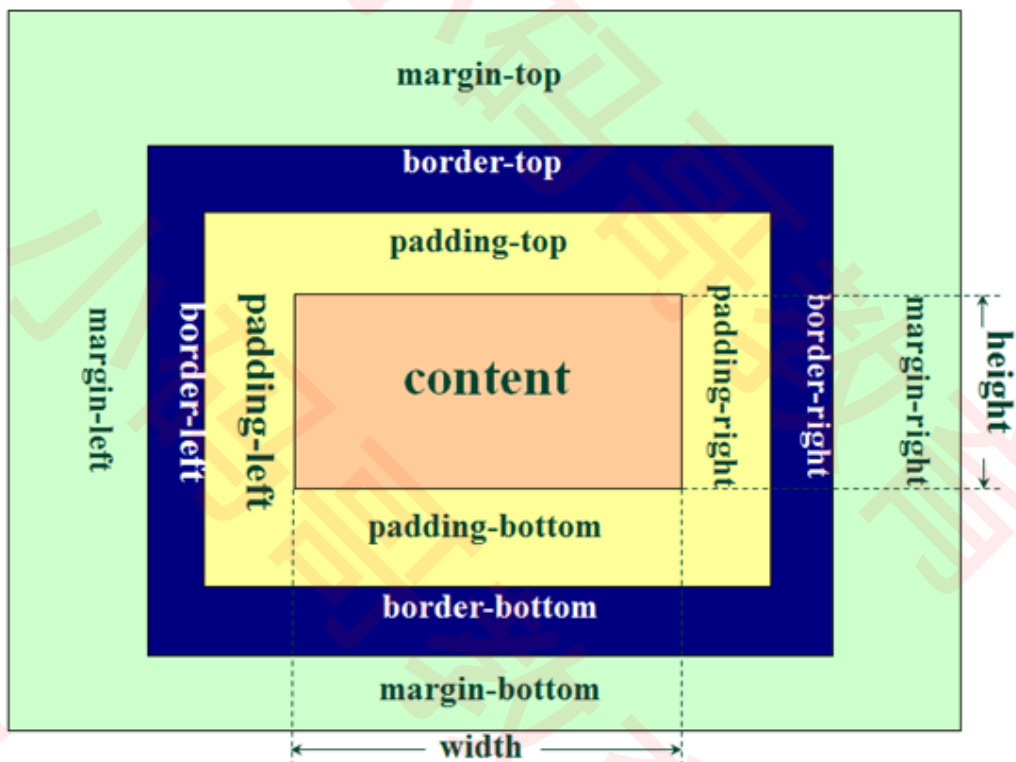
# box-sizing: border-box;



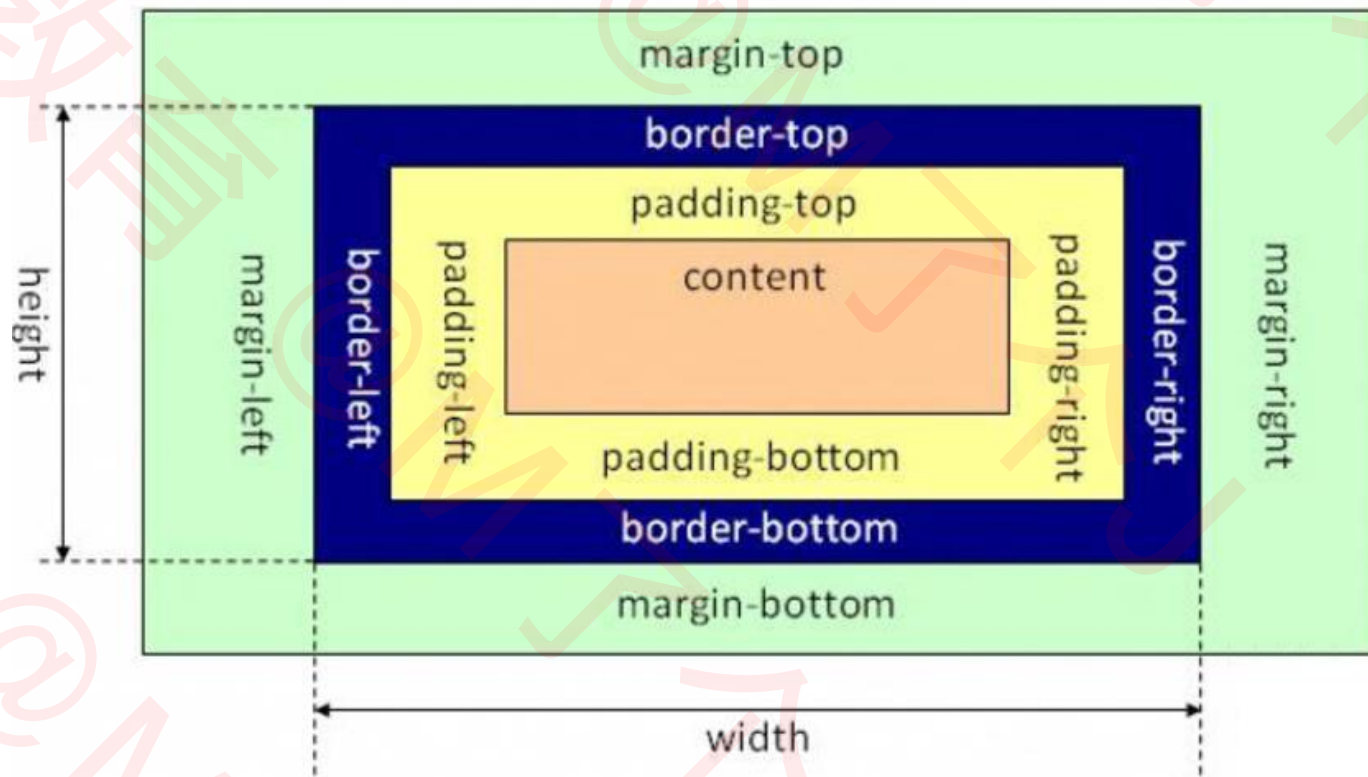
- 元素的实际占用宽度 = width
- 元素的实际占用高度 = height

# 盒子模型

W3C标准盒子模型



IE盒子模型 (IE8以下浏览器)



# 元素的水平居中显示

- 在一些需求中，需要元素在父元素中水平居中显示（父元素一般都是块级元素、inline-block）
- 行内级元素、inline-block元素
  - 水平居中：在父元素中设置text-align: center
- 块级元素
  - 水平居中：给自己设置margin-left: auto、margin-right: auto