

# 学习目标

- 能够说出盒子模型的基本含义和主要属性
- 能说出盒子有那几部分组成
- 能写出盒子边框综合写法
- 能说出内边距影响盒子大小的情况
- 能说出内边距后面跟几个数值表示的意思
- 能说出块级盒子如何居中对齐

# CSS三大模块

其实，CSS就三个大模块：**盒子模型**、**浮动**、**定位**，其余的都是细节。**要求这三部分，无论如何也要学的非常精通。**

**所谓盒子模型：**

就是把HTML页面中的布局元素看作是一个矩形的盒子，也就是一个盛装内容的容器。每个矩形都由元素的内容、内边距(padding)、边框(border)和外边距(margin)组成。

# 看透网页布局的本质

网页布局中，我们是如何把里面的文字，图片，按照美工给我们的效果图排列的整齐有序呢？



牛奶是怎样运输，让消费者购买的呢？



我们说过，行内元素比如 文字 类似牛奶，也需要一个盒子把他们装起来，我们前面学过的双标签都是一个盒子。有了盒子，我们就可以随意的，自由的，摆放位置了。



看透网页布局的本质:

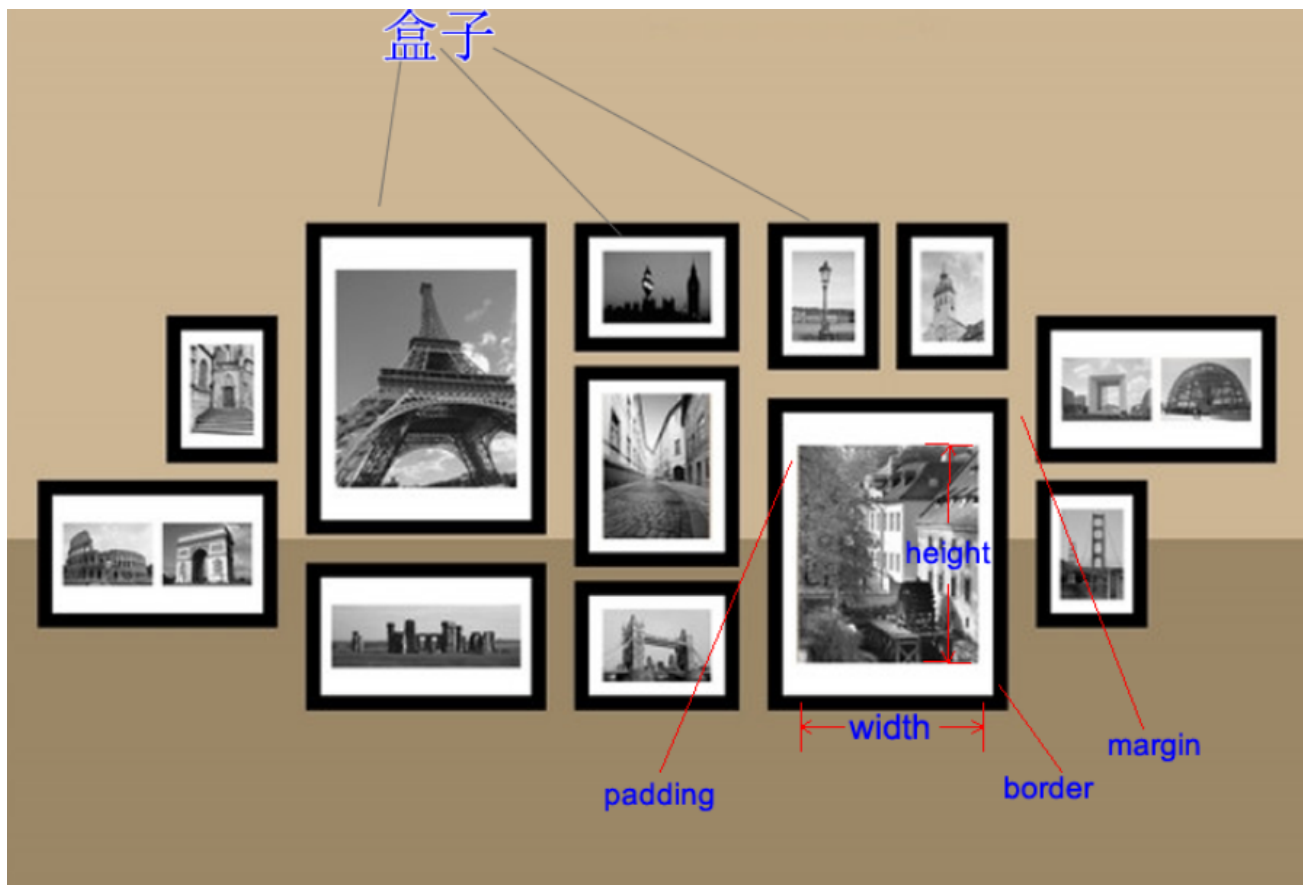
- 把网页元素比如文字图片等等，放入盒子里面。
- 然后利用CSS摆放盒子的过程，就是网页布局。
- 摆盒子的过程 就是网页布局



CSS 其实没有太多逻辑可言，类似我们小时候玩的积木,我们可以自由的，随意的摆放出我们想要的效果。



# 盒子模型(Box Model)



在CSS中一个盒子的组成部分:内容(content)+内填充(padding)+边框(border)+外边距(margin)

一个盒子中的主要属性:width、height、border、padding、margin

**width:** 指"宽度"的意思 但是这里的宽度指的盒子里面的内容的宽度 而不是盒子的宽度

**height:** 指"高度"的意思 但是这里的高度指的盒子里面的内容的高度 而不是盒子的高度

**border:** 是"外边框"的意思 指的盒子的边框

**padding:** 是"内填充"的意思, 指的盒子里面的内容到盒子边框的距离

**margin:** 是"外边距"的意思 指的是盒子与盒子之间的间距

## 盒子边框(border) 重要

盒子边框也就是CSS边框属性border

可以理解边框就是那层皮。橘子皮。。柚子皮。。橙子皮。。。

语法:

```
css选择器{  
    border : border-width border-style border-color  
}
```

属性解析:

**border-width:**设置边框的宽度,单位px

**border-style:**设置边框的风格，常用属性值如下:

属性值	描述
none(常用)	没有边框即忽略所有边框的宽度(默认值)
solid(常用)	边框为单实线(最为常用的)
dashed(常用)	边框为虚线
dotted	边框为点线
double	边框为双实线

**border-color:**设置边框的颜色;

**注意:**border-style不能省略,border-width默认3px,border-color默认黑色

举例:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
    <style>  
        /*border是设置盒子(元素,标签)的边框*/  
        /*语法*/  
        /*设置四个方向的边框*/  
        /*border:border-width border-style border-color;*/  
        /*border:边框粗细 边框类型 边框颜色*/  
        /*border-width取值数值,带px单位*/  
        /*border-style取值 solid实线 dashed虚线 dotted点状线 none没有线 double双实线 常用  
solid,none,dashed*/  
        /*border-color取值颜色值*/  
        .div1{  
            width: 200px;  
            height: 200px;  
            border:5px solid blue;  
            /*border:5px dashed blue;*/  
        }
```

```

        /*border:5px dotted blue;*/
        /*border:5px none blue;*/
        /*border:5px double blue;*/
    }

    .div2{
        width: 300px;
        height: 300px;
        background: orange;
        /*可以单独设置某个方向的边框 加个方向英文单词即可*/
        /*上边框*/
        /*border-top:border-width border-style border-color;*/
        border-top:5px solid green;

        /*下边框*/
        /*border-bottom:border-width border-style border-color;*/
        border-bottom:10px dashed blue;

        /*左边框*/
        /*border-left:border-width border-style border-color;*/
        border-left:20px dotted red;

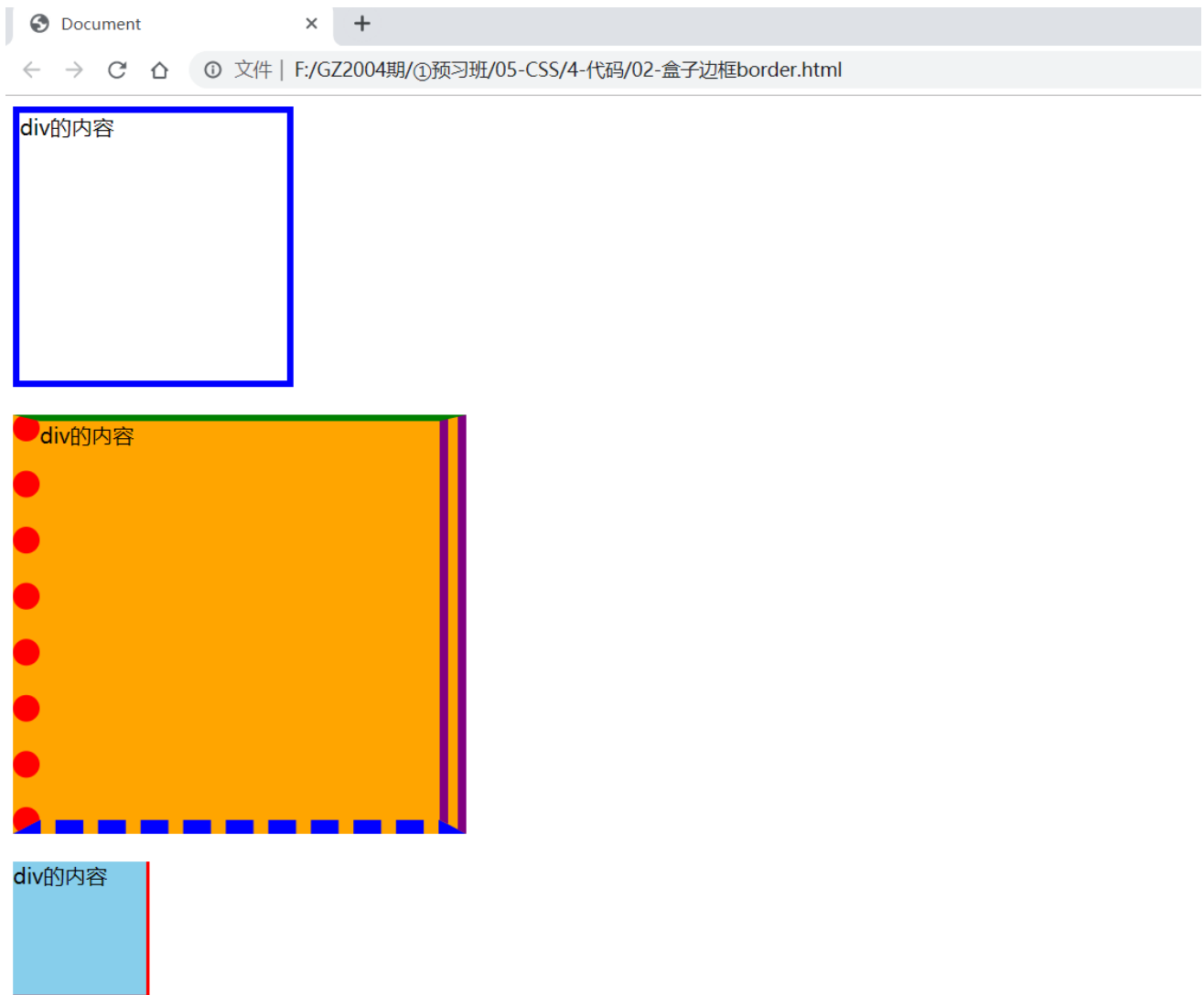
        /*右边框*/
        /*border-right:border-width border-style border-color;*/
        border-right:20px double purple;
    }

    .div3{
        width: 100px;
        height: 100px;
        background: skyblue;
        border:2px solid red;

        /*去掉左边框线*/
        /*border-left:2px none red;*/
        border-left:none;

        /*去掉上边框线*/
        border-top:none;
    }
</style>
</head>
<body>
    <div class="div1">div的内容</div>
    <br/>
    <div class="div2">div的内容</div>
    <br/>
    <div class="div3">div的内容</div>
</body>
</html>

```



## 圆角边框(CSS3)

语法格式:

```
css选择器{  
    /*一个值,设置元素4个角*/  
    border-radius:长度值或百分比;  
  
    /*border-radius设置4个值,可以设置元素的4个角*/  
    border-radius: 左上角 右上角 右下角 左下角; 顺时针的方向;  
}
```

注意: border-radius : 50%; 可以让一个正方形 变成圆 原理就是 原点到半径相同, 就可以画一个圆圈。



## 举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .login{
      width: 100px;
      height: 50px;
      background: orange;
      text-align: center;
      line-height: 50px;
      color:white;
      /*border:3px solid red;*/
      /*圆角边框*/
      /*设置四个角*/
      /*border-radius: 一个数值带px或者一个百分比,没有负值*/
      border-radius: 10px;
    }
    .circle{
      width: 200px;
      height: 200px;
      background: pink;
      /*设置正方形的元素的border-radius:50%,就可以变成一个圆*/
      border-radius:50%;
      /*border-radius: 10%;*/
      text-align: center;
      line-height: 200px;
    }
    .changfangxing{
      width: 200px;
      height: 50px;
      border-radius:50%;
      background: skyblue;
      text-align: center;
      line-height: 50px;
    }
    img{
      border-radius: 50%;
    }

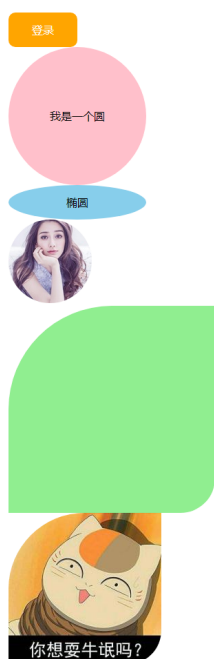
    .div1{
      width: 300px;
      height: 300px;
      background: lightgreen;
      /*border-radius设置4个值*/
      /*border-radius: 左上角 右上角 右下角 左下角; 顺时针的方向*/
      border-radius: 50% 0px 50px 0px;
    }
  </style>
</head>
<body>
  <div class="login">
    登录
  </div>
  <div class="circle">
    圆
  </div>
  <div class="changfangxing">
    长方形
  </div>
  <img alt="圆角矩形" data-bbox="104 137 667 920"/>
  <div class="div1">
    圆角矩形
  </div>
</body>
</html>
```



```

        .img2{
            border-radius: 50% 0px 50px 0px;
        }
    </style>
</head>
<body>
    <div class="login">登录</div>
    <div class="circle">我是一个圆</div>
    <div class="changfangxing">椭圆</div>
    
    <div class="div1"></div>
    
</body>
</html>

```



## 表格的细线边框

用table的rules="all"也可以实现更细的边框,但是这个有一定的兼容问题,更推荐使用css来合并边框线!

以前学过的html表格边框很粗, 这里只需要CSS一句话就可以美观起来。 让我们真的相信, CSS就是我们的白马王子(白雪公主)。

**语法:**

```
table{ border-collapse:collapse; }
```

说明:

collapse 单词是合并的意思

border-collapse:collapse; 表示相邻边框合并在一起。

课堂案例:

## 小说排行榜

排名	关键词	趋势	今日搜索	最近七日	相关链接
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    tr:nth-child(odd){
      background: #ffc0cb;
    }
    tr:first-child{
      background: #ff69b4;
    }
    table{
      width: 500px;
      border:1px solid #ff3ba5;
      /*表格更细的边框线,折叠边框线,需要加到table标签上*/
      border-collapse: collapse;
    }
    table tr td{
      border:1px solid #ff3ba5;
    }
    table tr th{
      border:1px solid #ff3ba5;
    }
  </style>
</head>
<body>
  <table>
```

```
<caption><h3>小说排行榜</h3></caption>
<tr>
  <th>排名</th>
  <th>关键词</th>
  <th>趋势</th>
  <th>今日搜索</th>
  <th>最近七日</th>
  <th>相关链接</th>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
  <td>356</td>
  <td>3560</td>
  <td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
  <td>1</td>
  <td>鬼吹灯</td>
  <td>1</td>
```

```
<td>356</td>
<td>3560</td>
<td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
<tr>
<td>1</td>
<td>鬼吹灯</td>
<td>1</td>
<td>356</td>
<td>3560</td>
<td><a href="#">贴吧</a><a href="#">图片</a><a href="#">百科</a></td>
</tr>
</table>
</body>
</html>
```

# 内填充(padding) 重要

## padding基本使用

padding属性用于设置内边距。是指 边框与内容之间的距离。

方法有两种:第一种我们称之为**全写属性**, 第二种我们称之为**简写属性**

**全写属性:**

- **padding-top:**上内边距
- **padding-right:**右内边距
- **padding-bottom:**下内边距
- **padding-left:**左内边距

**简写属性:** **\*\*padding\*\***:这个属性是有方向的 可以同时表示四个方向 这个属性的方向是有顺序的 顺序是顺时针方向 也就是:上、右、下、左 这个顺序

**注意:**后面跟几个数值表示的意思是不一样的

值的个数	表达意思
1个值	padding:上下左右边距 比如padding: 3px; 表示上下左右都是3像素
2个值	padding: 上下边距 左右边距 比如 padding: 3px 5px; 表示 上下3像素 左右 5像素
3个值	padding:上边距 左右边距 下边距 比如 padding: 3px 5px 10px; 表示 上是3像素 左右是5像素 下是10像素
4个值	padding:上内边距 右内边距 下内边距 左内边距 比如: padding: 3px 5px 10px 15px; 表示 上3px 右是5px 下 10px 左15px 顺时针

### 举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*padding内填充,指得是内容到边框的距离*/
    /*padding-top上填充*/
    /*padding-bottom下填充*/
    /*padding-left左填充*/
    /*padding-right右填充*/
    /*padding取值数值带px单位,只能正值*/

    .div1{
      width: 300px;
      height: 30px;

      background: pink;
      border:2px solid blue;
      padding-top: 10px;
      padding-bottom: 20px;
      padding-left: 50px;

      text-align: right;
      padding-right: 10px;
    }
    .div2{
      width: 100px;
      height: 100px;
      background: skyblue;

      /*padding简写方式*/
      /*padding:一个值; 上下左右都是这个值*/
      /*padding:10px;*/

      /*padding:两个值; 第一个值是上下填充,第二个值是左右填充*/
```

```

        /*padding:10px 20px;*/

        /*padding:三个值;第一个值上填充,第二个值左右填充,第三个下填充*/
        /*padding:10px 20px 30px;*/

        /*padding:四个值; 上右下左,顺时针*/
        padding:10px 20px 30px 40px;
    }
</style>
</head>
<body>
    <div class="div1">今天星期五,明天我休息</div>
    <br/>
    <div class="div2">明天你们要自习,哈哈哈</div>
</body>
</html>

```

今天星期五,明天我休息

明天你们要自  
习,哈哈哈

## padding会撑开盒子宽度跟高度

因为padding会撑开盒子宽度跟高度,所以我们得知道如何**计算盒子的真实宽度与高度**

### 盒子的实际大小

盒子的实际的大小 = **内容的宽度和高度 + 内边距 + 边框**

计算盒子的真实宽度 = 盒子里面内容的宽度 + 左右两边的内填充 + 左右两边的边框线

计算盒子的真实高度 = 盒子里面内容的高度 + 上下两边的内填充 + 上下两边的边框线

这里我们要严格遵循 效果图里面的大小，不能随意的变大，超出范围，否则会影响 其余盒子的布局。

所以我们要学会计算盒子的大小 以保证 我们内容的大小的 在正确的范围。

**举例:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: pink;
      border-top:1px solid red;
      border-bottom:2px dashed blue;
      border-left:5px dotted orange;
      border-right: 10px double green;

      padding-top: 10px;
      padding-bottom: 20px;
      padding-left: 30px;
      padding-right: 40px;

      /*padding会撑开盒子宽度跟高度*/
      /*盒子的真实宽度 = width+左填充+右填充+左边框+右边框*/
      /*盒子的真实宽度 = 100+30+40+5+10 = 185*/

      /*盒子的真实高度 = height+上填充+下填充+上边框+下边框*/
      /*盒子的真实高度 = 100+10+20+1+2 = 133*/
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

**保证盒子真实宽度不变:** 高度也同理

如果增加了内填充就要减去盒子里面内容的宽度

如果增加了内容的宽度减去盒子里面的内填充的宽度

**举例:**

```
<!DOCTYPE html>
```



```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    div{
      background: pink;
      border-top:1px solid red;
      border-bottom:2px dashed blue;
      border-left:5px dotted orange;
      border-right: 10px double green;

      padding-top: 10px;
      padding-bottom: 20px;
      padding-left: 30px;
      padding-right: 40px;

      /*padding会撑开盒子宽度跟高度*/
      /*盒子的真实宽度 = width+左填充+右填充+左边框+右边框*/
      /*盒子的真实宽度 = 100+30+40+5+10 = 185*/

      /*盒子的真实高度 = height+上填充+下填充+上边框+下边框*/
      /*盒子的真实高度 = 100+10+20+1+2 = 133*/

      /*如何保证盒子真实宽度或者真实高度不变 多了多少,就减去多少width或者height值*/
      /*width: 100px;*/
      width: 15px;

      /*height: 100px;*/
      height: 67px;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>

```

### 练习:

页面有一个400\*400的盒子，盒子有一个10px的上边框和左边框，有一个20px的下内边距，请问盒子的width和height应该设置成大？  
390宽度 370高度

### 总结:

border和padding会影响盒子的大小, margin不会  
盒子的真实宽度: width+左右border+左右padding  
盒子的真实高度: height+上下border+上下padding

## padding不影响盒子宽度情况

如果没有给一个"块级"元素指定宽度, 默认宽度是相对父元素的100% 此时, 如果给这个盒子指定左右padding, 则不会撑开盒子。

举例:

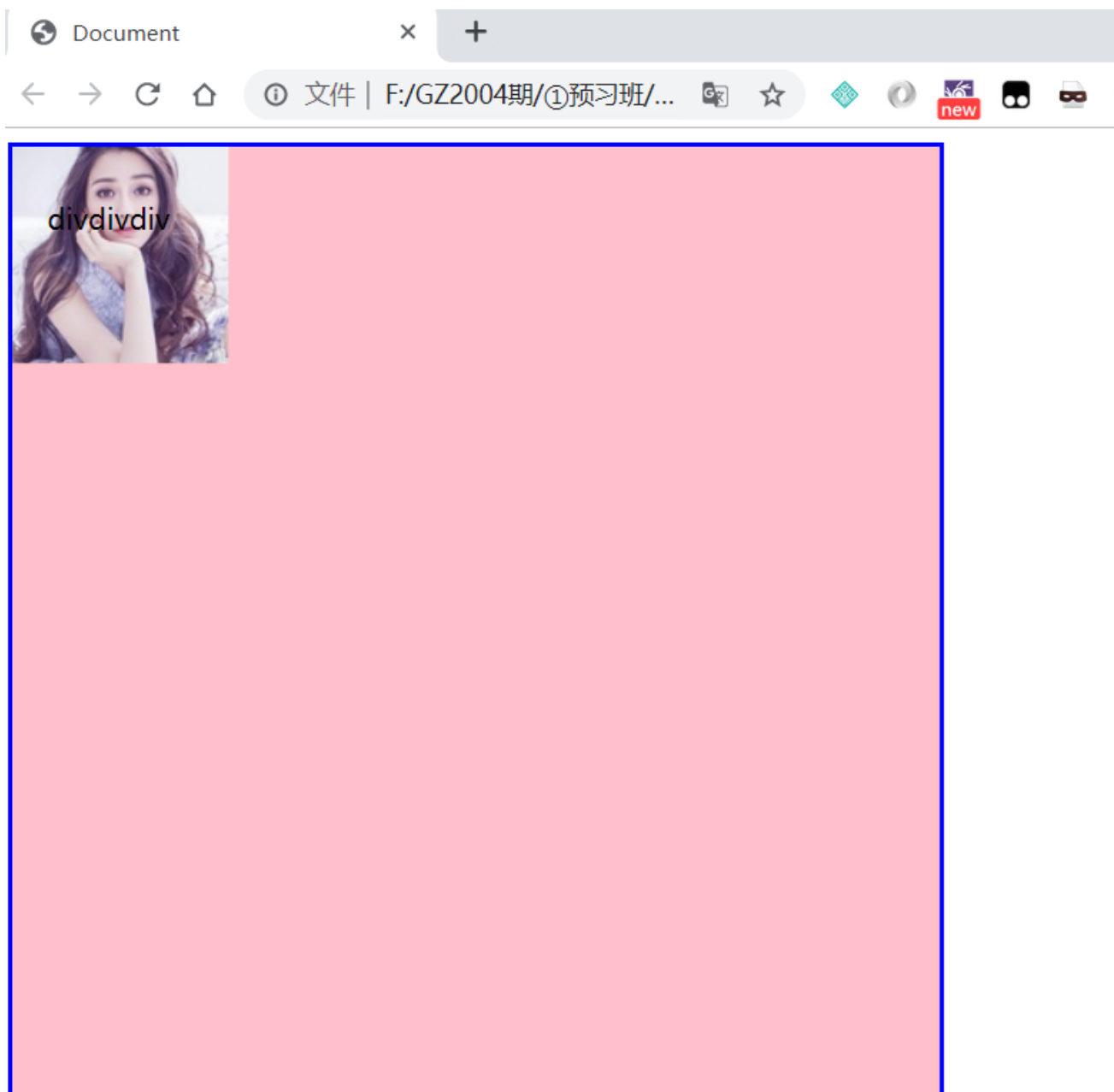
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*padding不影响盒子宽度情况*/
    /*
      如果没有给一个"块级"元素指定宽度, 默认宽度是相对父元素的100% 此时, 如果给这个盒子指定左
      右padding, 则不会撑开盒子。
    */
    p{
      background: skyblue;
      /*左填充*/
      padding-left: 100px;
      /*右填充*/
      padding-right: 200px;
    }
  </style>
</head>
<body>
  <p>p段落内容</p>
</body>
</html>
```

## 给元素设置背景图片与背景颜色是不受padding影响的

padding只能影响元素里面的内容,比如文字内容

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*给元素设置背景图片与背景颜色是不受padding影响的*/
    /*padding只影响元素中的内容*/
    div{
      width: 500px;
      height: 500px;
      background: pink url(images/1.jpg) no-repeat;
      border:2px solid blue;
      padding-left:20px;
      padding-top: 30px;
    }
  </style>
</head>
<body>
  <div>divdivdiv</div>
</body>
</html>
```



## 课堂测试

1. 关于盒子真实宽度下列正确的是()  
(A) 盒子宽:就是width的大小 (B) 盒子宽: padding-left + width + padding-right (C) 盒子宽: border-left + width + border-right (D) 盒子宽: border-left+ padding-left + width + padding-right + border-right
2. 一个盒子宽度为100，padding为10，边框为5像素，问这个盒子实际的宽度的的是() (A) 130 (B) 135 (C) 125 (D) 115
3. 关于根据下列代码计算 盒子宽高下列说法正确的是()

```
<!DOCTYPE html>
```

```






<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style type="text/css">
    div {
      width: 200px;
      height: 200px;
      border: 1px solid #000000;
      border-top: 5px solid blue;
      padding: 50px;
      padding-left: 100px;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>

```

(A) 宽度为200px 高度为200px (B) 宽度为352px 高度为306px (C) 宽度为302px 高度为307px (D) 宽度为302px 高度为252px

## 课堂练习

要求使用ul li a标签完成, 并且背景样式完成以下效果

 大明星:姜潮魔性拜年道晚安  
 软萌正太徐浩演绎《小幸运》  
 漫威绝逼好看的电影镜头合集  
 从没见过这么搞笑的祖孙组合  
 史上最容易挨揍的自助餐吃法



```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    a{
      text-decoration: none;
    }
  </style>

```

```

    }
    a:hover{
        text-decoration: underline;
        color: pink;
    }
    li{
        list-style: none;
        background: url(images/li.gif) no-repeat;
        /*想看一个元素的位置,可以给这个元素设置背景颜色或者边框线*/
        /*border: 1px solid red;*/
        /*text-indent: 25px;*/
        padding-left: 25px;
        /*想要修改背景图片的位置,只能使用背景定位*/
        background-position: 0px center;
    }
</style>
</head>
<body>
    <ul>
        <li><a href="#">大明星: 姜潮魔性拜年道晚安</a></li>
        <li><a href="#">软萌正太徐浩演绎《小幸运》</a></li>
        <li><a href="#">漫威绝逼好看的电影镜头合集</a></li>
        <li><a href="#">从没见过这么搞笑的祖孙组合</a></li>
        <li><a href="#">史上最容易挨揍的自助餐吃法</a></li>
    </ul>
</body>
</html>

```

# 外边距(margin) 重要

## margin基本使用

margin属性用于设置外边距。margin就是控制盒子和盒子之间的距离

- margin-top: 上外边距
- margin-right: 右外边距
- margin-bottom: 下外边距
- margin-left: 左外边距

margin: 上外边距 右外边距 下外边距 左外边距

取值顺序跟内边距相同。

举例:

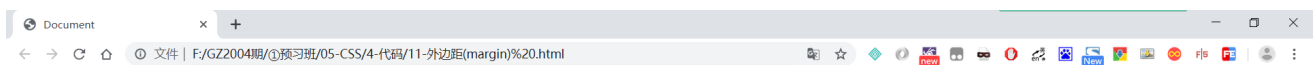
```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*margin指的盒子与盒子之间的距离*/
    /*margin-top上外边距*/
    /*margin-bottom下外边距*/
    /*margin-left左外边距*/
    /*margin-right右外边距*/
    /*取值:数值带px单位,可以正值也可以负值,还有个auto的取值,代表自动*/
    .one{
      width: 100px;
      height: 100px;
      background: red;
      margin-left: 50px;
      margin-left: -20px;

      margin-top: 50px;
      /*margin-bottom: 50px;*/
    }
    .two{
      width: 100px;
      height: 100px;
      background: blue;
      /*右浮*/
      float: right;
      margin-right: 100px;
    }
  </style>
</head>
<body>
  <div class="one">one</div>
  <div class="two">two</div>
</body>
</html>

```



**margin实现盒子水平居中(重点,常用,必须得掌握)**



可以让一个盒子实现水平居中，需要满足一下两个条件：

1. 必须是块级元素。
2. 盒子必须指定了小于父元素100%的宽度(width)
3. 然后就给左右的外边距都设置为auto，就可使块级元素水平居中。

实际工作中常用这种方式进行网页布局，示例代码如下：

```
.header{ width:960px; margin:0 auto;}
```

举例：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .div1{
      width: 500px;
      height: 100px;
      background: red;
      text-align:center;
      /*
        想让一个盒子居中,有3个条件
        第一:需要是一个块级元素
        第二:需要给这个块级元素设置宽度,并且宽度要小于100%
        第三:再设置margin-left:auto;以及margin-right:auto;

        auto自动的意思,只对左外边距以及右外边距有效
      */
      /*margin-left: auto;*/
      /*margin-right: auto;*/

      /*推荐使用这种*/
      margin:0 auto;

      /*margin:auto;*/
    }
  </style>
</head>
<body>
  <div class="div1">div的内容</div>
</body>
</html>
```

# 文字水平居中,盒子水平居中,图片水平居中,背景图片水平居中的区别

1. 文字水平居中是 text-align: center
2. 盒子水平居中 左右margin 改为 auto

```
text-align: center; /* 文字居中水平 */  
margin: 10px auto; /* 盒子水平居中 左右margin设置auto就阔以了 上下margin的值没要求*/
```

3. 图片水平居中
4. 背景水平居中

举例:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Document</title>  
  <style>  
    .div1{  
      width: 500px;  
      height: 100px;  
      background: red;  
      text-align:center;  
    }  
    .div2{  
      width: 500px;  
      height: 100px;  
      background: blue;  
      margin:0 auto;  
    }  
    .div3{  
      width: 500px;  
      height: 300px;  
      background: orange;  
      text-align: center;  
    }  
    .div4{  
      /*width: 500px;*/  
      width: 310px;  
      height: 300px;  
      background: skyblue;  
      /*左填充*/  
      padding-left: 190px;  
    }  
    .div5{  
      width: 500px;  
      height: 300px;
```



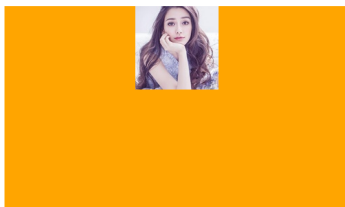
文字水平居中 text-align:center



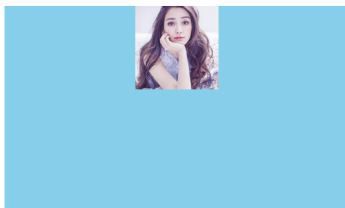
盒子水平居中 margin-left:auto; margin-right:auto;



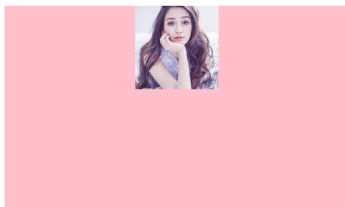
图片水平居中 第一种方法:给父元素设置text-align:center



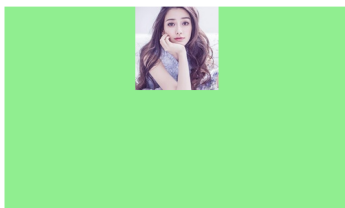
图片水平居中 第二种方法:给父元素设置padding-left



图片水平居中 第三种方法:给图片元素设置margin-left



背景图片水平居中中的区别



## 清除页面所有元素的默认内外边距

为了方便地控制网页中的元素，制作网页时，可使用如下代码清除元素的默认内外边距:

```

* {
  padding:0;          /* 清除内边距 */
  margin:0;           /* 清除外边距 */
}

```

也可以这样做

```

/*分组选择器*/
html,body,h1,h2,h3,h4,h5,h6,div,span,p,ul,li,dl,dt,dd,img,input,ol{
  margin: 0;
  padding:0;
}

```

## 举例:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    body{
      /*margin:0px;*/
    }
    .div1{
      background: red;
      width: 100%;
    }
    h1{
      background: blue;
      /*margin: 0;*/
    }
    h2{
      background: orange;
      /*margin: 0;*/
    }
    /*ul{
      margin: 0;
      padding: 0;
    }*/
    /*因为HTML标签中,很多标签自带margin外边距或者padding内填充,会影响布局,所以我们喜欢在布局的时候,把所有HTML标签的内外边距都清0*/
    *{
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>
  <div class="div1">div1</div>
  <h1>标题1标签</h1>

```

```
<h2>标题2标签</h2>
<ul>
  <li>苹果</li>
  <li>西瓜</li>
  <li>木瓜</li>
  <li>傻瓜</li>
</ul>
</body>
</html>
```

## 行内元素为了照顾兼容性,尽量只设置左右内外边距,不要设置上下内外边距

注意:默认情况下,行内元素为了照顾兼容性, 尽量只设置左右内外边距, 不要设置上下内外边距。默认情况下,行内元素设置上下的内填充与外边距有问题

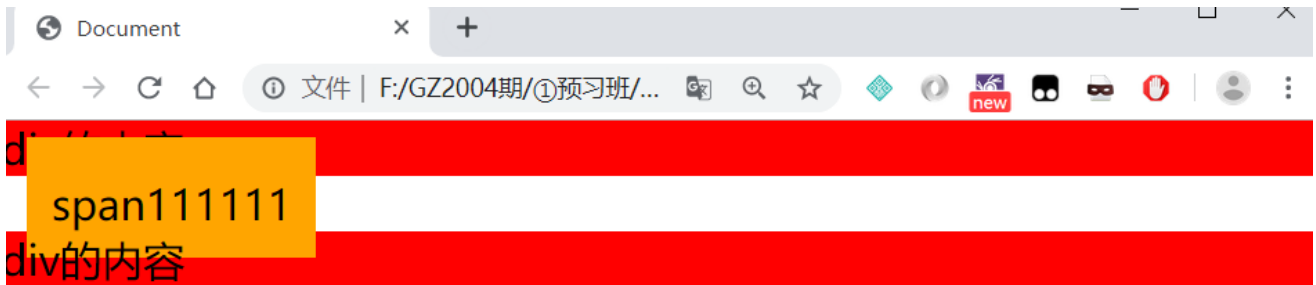
举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    div{
      background: red;
    }
    span{
      background: orange;
      padding-left: 10px;
      padding-right: 10px;
      padding-top: 15px;
      padding-bottom: 10px;

      margin-left: 10px;
      margin-right: 20px;
      margin-top: 100px;
      margin-bottom: 50px;

      /*行内元素为了照顾兼容性,尽量只设置左右内外边距,不要设置上下内外边距*/
    }
  </style>
</head>
<body>
```

```
<div>div的内容</div>
<span>span111111</span>
<div>div的内容</div>
</body>
</html>
```



## padding与margin设置负值问题

结论 padding设置负值无效,margin设置负值有效

举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*padding与margin设置负值问题*/
    /*margin可以设置正值,也设置负值,负值有效*/
    /*padding只可以设置正值,负值无效*/
    *{
      margin: 0;
      padding: 0;
    }
    div{
      width: 100px;
      height: 100px;
      background:red;
      margin-left: -50px;
      padding-left: -20px;
    }
  </style>
</head>
<body>
  <div>div的内容</div>
</body>
</html>
```

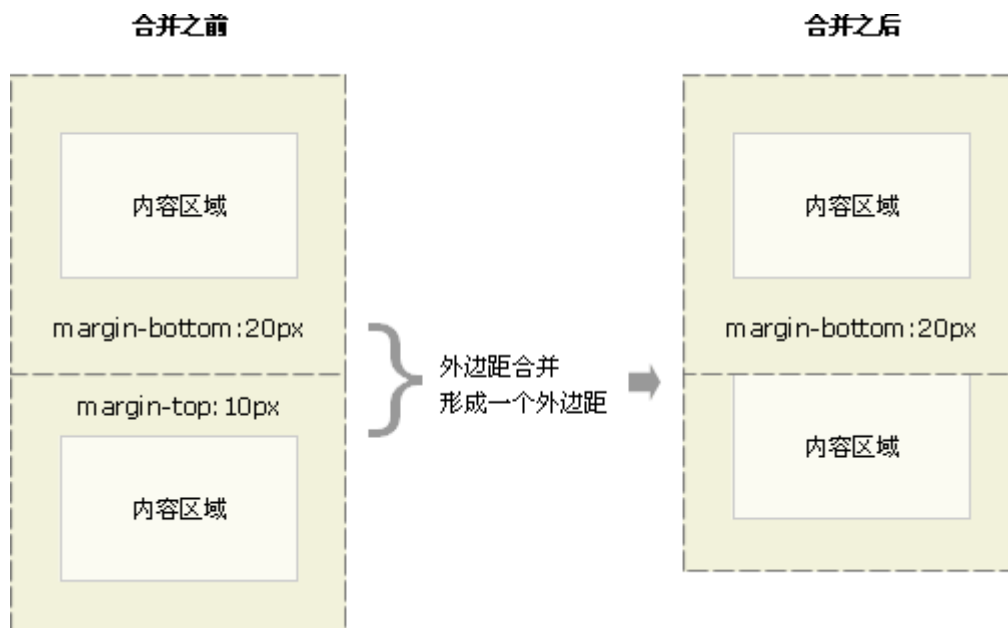


## margin外边距合并现象(margin常见问题)

使用margin定义块元素的垂直外边距时，可能会出现外边距的合并。

### 相邻(兄弟关系)块元素垂直外边距的合并

当上下相邻的两个块元素相遇时，如果上面的元素有下外边距margin-bottom，下面的元素有上外边距margin-top，则他们之间的垂直间距不是margin-bottom与margin-top之和，而是两者中的**较大者**。这种现象被称为相邻块元素垂直外边距的合并(也称外边距塌陷)。



**解决方案:**如果是垂直方向的盒子在设置外边距时 只需要设置一个就好

**举例:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>
  <div>
    <div>内容区域</div>
    <div>margin-bottom: 20px</div>
    <div>margin-top: 10px</div>
    <div>内容区域</div>
  </div>
</body>
</html>
```

/\*相邻(兄弟关系)块元素垂直方向上外边距会出现合并现象\*/

/\*当上下相邻的两个块元素相遇时，如果上面的元素有下外边距margin-bottom，下面的元素有上外边距margin-top，则他们之间的垂直间距不是margin-bottom与margin-top之和，而是两者中的较大者。这种现象被称为相邻块元素垂直外边距的合并(也称外边距塌陷)。\*/

```

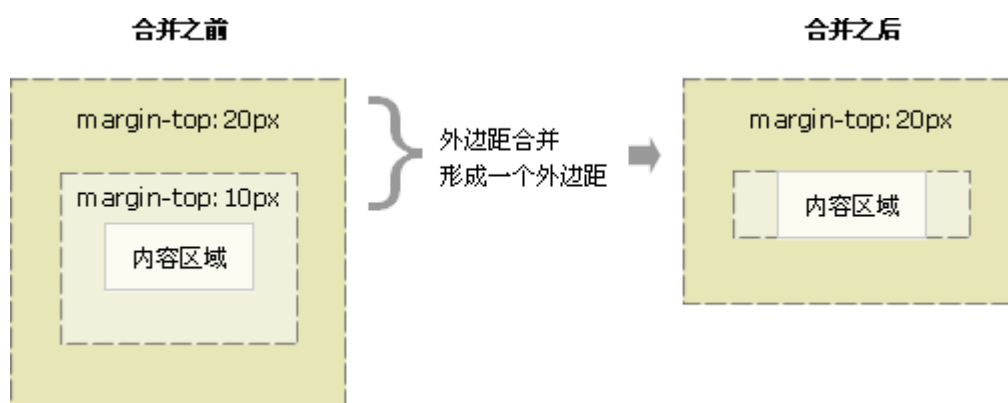
.one{
  width: 100px;
  height: 100px;
  background: red;
  /*上面的盒子设置下外边距*/
  /*margin-bottom: 20px;*/

  /*解决方案:如果是垂直方向的盒子在设置外边距时 只需要设置一个就好*/
  margin-bottom: 30px;
}
.two{
  width: 100px;
  height: 100px;
  background: blue;
  /*下面的盒子设置上外边距*/
  /*margin-top: 10px;*/
}
</style>
</head>
<body>
  <div class="one">one</div>
  <div class="two">two</div>
</body>
</html>

```

## 嵌套(父子关系)块元素垂直外边距的合并

对于两个嵌套关系的块元素，如果父元素没有上内边距及边框，则父元素的上外边距会与子元素的上外边距发生合并，合并后的外边距为两者中的较大者，即使父元素的上外边距为0，也会发生合并。



解决方案:

1. 可以为父元素定义1像素的上边框或上内边距。
2. 可以为父元素添加overflow:hidden。

举例:

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    /*嵌套(父子关系)块元素垂直外边距的合并*/
    /*对于两个嵌套关系的块元素，如果父元素没有上内边距及边框，则父元素的上外边距会与子元素的上外边距
    发生合并，合并后的外边距为两者中的
    较大者”，即使父元素的上外边距为0，也会发生合并。*/
    .father{
      width: 500px;
      height: 500px;
      background: red;
      margin-top: 20px;
      /*解决方案:*/
      /*1. 可以为父元素定义1像素的上边框或上内边距。*/
      /*border-top:1px solid orange;*/
      /*padding-top: 1px;*/

      /*2. 可以为父元素添加overflow:hidden 推荐使用*/
      overflow: hidden;
    }
    .father .son{
      width: 100px;
      height: 100px;
      background: blue;
      margin-top: 10px;
    }
  </style>
</head>
<body>
  <div class="father">
    <div class="son"></div>
  </div>
</body>
</html>

```

## 嵌套块(父子关系)元素给子元素设置上外边距出现整体塌陷(掉下来)现象

解决方法:

1. 可以为父元素定义1像素的上边框或上内边距
2. 可以为父元素添加overflow:hidden
3. 也可以给父元素设置等价的padding-top,记得减去对应的高度,也可以解决

举例:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    /*嵌套块(父子关系)元素给子元素设置上外边距出现整体塌陷(掉下来)现象*/
    .father{
      width: 500px;
      height: 500px;
      background: red;
      /*
      解决方法:
        1. 可以为父元素定义1像素的上边框或上内边距
        2. 可以为父元素添加overflow:hidden 推荐使用
        3. 也可以给父元素设置等价的padding-top,记得减去对应的高度,也可以解决
      */
      /*border-top:1px solid orange;*/
      /*padding-top:1px;*/

      overflow: hidden;

      /*padding-top: 100px;*/
      /*height: 400px;*/
    }
    .father .son{
      width: 100px;
      height: 100px;
      background: blue;
      margin-top: 100px;
    }
  </style>
</head>
<body>
  <div class="father">
    <div class="son"></div>
  </div>
</body>
</html>

```

## 盒子模型布局稳定性

---

开始学习盒子模型，同学们最大的困惑就是，**分不清内外边距的使用，什么情况下使用内边距，什么情况下使用外边距？**

答案是: 其实他们大部分情况下是**可以混用的**。就是说，你用内边距也可以，用外边距也可以。你觉得哪个方便，就用哪个。

但是，总有一个最好用的吧，我们根据稳定性来分，建议如下：

按照 优先使用 宽度 (width) 其次 使用内边距(padding) 再次 外边距(margin)。

```
width > padding > margin
```

原因：

1. **margin 会有外边距合并 还有 ie6下面margin 加倍的bug(讨厌)所以最后使用。**
2. **padding 会影响盒子大小， 需要进行加减计算(麻烦) 其次使用。**
3. **width 没有问题我们经常使用宽度剩余法 高度剩余法来做。**

# 盒子阴影(CSS3)

盒子阴影是 CSS3当中新增的一个用于给盒子添加阴影的属性，属性名称叫 **box-shadow**

属性语法：

```
css选择器{
    /*box-shadow: 水平阴影的位置 垂直阴影的位置 模糊距离 阴影的尺寸 阴影的颜色 阴影类型;*/
    box-shadow: h-shadow v-shadow blur spread color inset;
}
```

属性说明：

值	描述	说明
h-shadow	必需。水平阴影的位置。允许负值。	单位 px,水平向右为正
v-shadow	必需。垂直阴影的位置。允许负值。	单位 px,垂直向下为正
blur	可选。模糊距离。	单位 px
spread	可选。阴影的尺寸。	单位 px
color	可选。阴影的颜色	颜色值
inset	可选。将外部阴影 (outset) 改为内部阴影(inset)。	默认不设置就是外部阴影; 只能设置为inset; 不能写outset;

注意：

- 可以实现给一个盒子加多个盒子阴影,多个阴影之间用 逗号 隔开
- 前两个属性是必须写的。其余的可以省略。
- 如果设置盒子阴影，默认就是外阴影 (outset)，不需要设置，设置了反而是错误的

#### 举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    .div1{
      width: 100px;
      height: 100px;
      background: red;
      margin:100px auto;
      /*box-shadow: 水平位置(正负数值px) 垂直位置(正负数值px) 阴影模糊程度(正数值px) 阴影大小
      阴影颜色;*/
      box-shadow: 0 15px 20px rgba(0,0,0,0.5);
    }
    .div2{
      width: 200px;
      height: 200px;
      border:2px solid blue;
      margin:0 auto;
      /*外部阴影*/
      /*box-shadow: 4px 6px 11px 1px green;*/
      /*内部阴影*/
      box-shadow: 4px 6px 11px 1px green inset;
    }
    .div3{
      width: 200px;
      height: 200px;
      background: pink;
      margin:50px auto;
      /*多阴影,可以给一个盒子设置多组阴影,每个阴影使用逗号隔开*/
      box-shadow: 1px 1px 1px 1px red , 4px 4px 1px 1px orange ,8px 8px 1px 1px
      yellow,12px 12px 1px 1px green;
    }
  </style>
</head>
<body>
  <div class="div1"></div>
  <div class="div2"></div>
  <div class="div3"></div>
</body>
</html>
```

# 综合案例

## 最新文章

### 最新文章/New Articles

北京招聘网页设计，平面设计，php

体验javascript的魅力

jquery世界来临

网页设计师的梦想

jquery中的链式编程是什么

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*第二步,先写初始化样式*/

    /*初始化样式开始 这段代码需要理解,需要熟悉*/
    /*清除所有标签的内外边距*/
    *{
      margin: 0;
      padding: 0;
    }
    /*去掉列表项目符号*/
    li{
      list-style:none;
    }
    /*超链接美化*/
    a{
      text-decoration: none;
      color:#202020;
```





```
</body>
</html>
```

## 今日总结

盒子模型由哪些部分组成? 内容width,height 边框border 内容padding 外边距margin

四个方向的边框 border:border-width border-style border-color;

border-top上边框

border-bottom下边框

border-left左边框

border-right右边框

圆角边框 border-radius:数值带px或者百分比; 一个值 或者 四个值

border-radius:50%;可以让正方形变成一个圆形

border-collapse:collapse;可以合并表格边框线,要给在table标签设置这个样式

padding内容到边框的距离

padding-top

padding-bottom

padding-left

padding-right

padding:10px;

padding:20px 50px;

padding:10px 20px 40px;

padding:10px 20px 30px 40px;

padding会撑开盒子宽度跟高度

盒子的真实宽度 = width+左边框+右边框+左填充+右填充

盒子的真实高度 = height+上边框+下边框+上填充+下填充

如何保证加了边框以及填充以后,盒子的真实宽度高度不变? 加了多少就减去相应多少的宽度或者高度