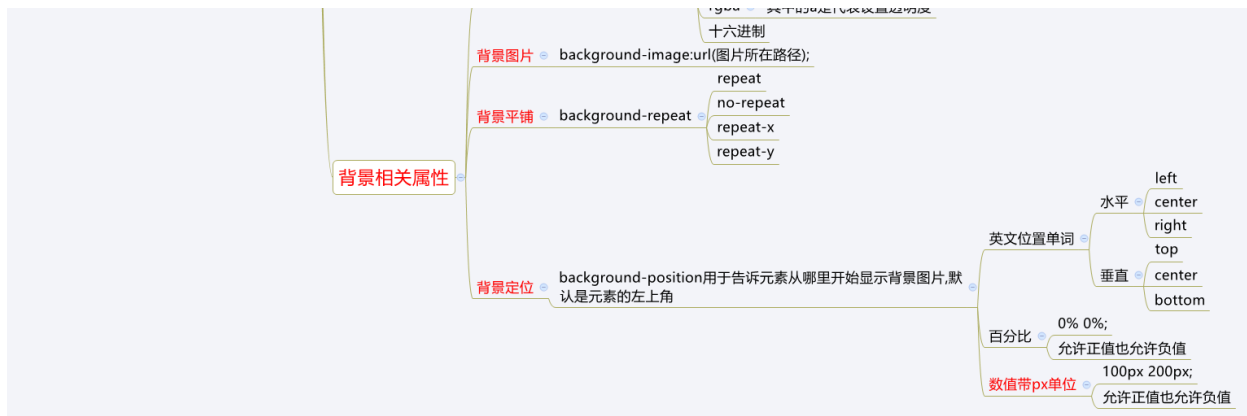


# 昨日回顾

## CSS第二天知识点总结





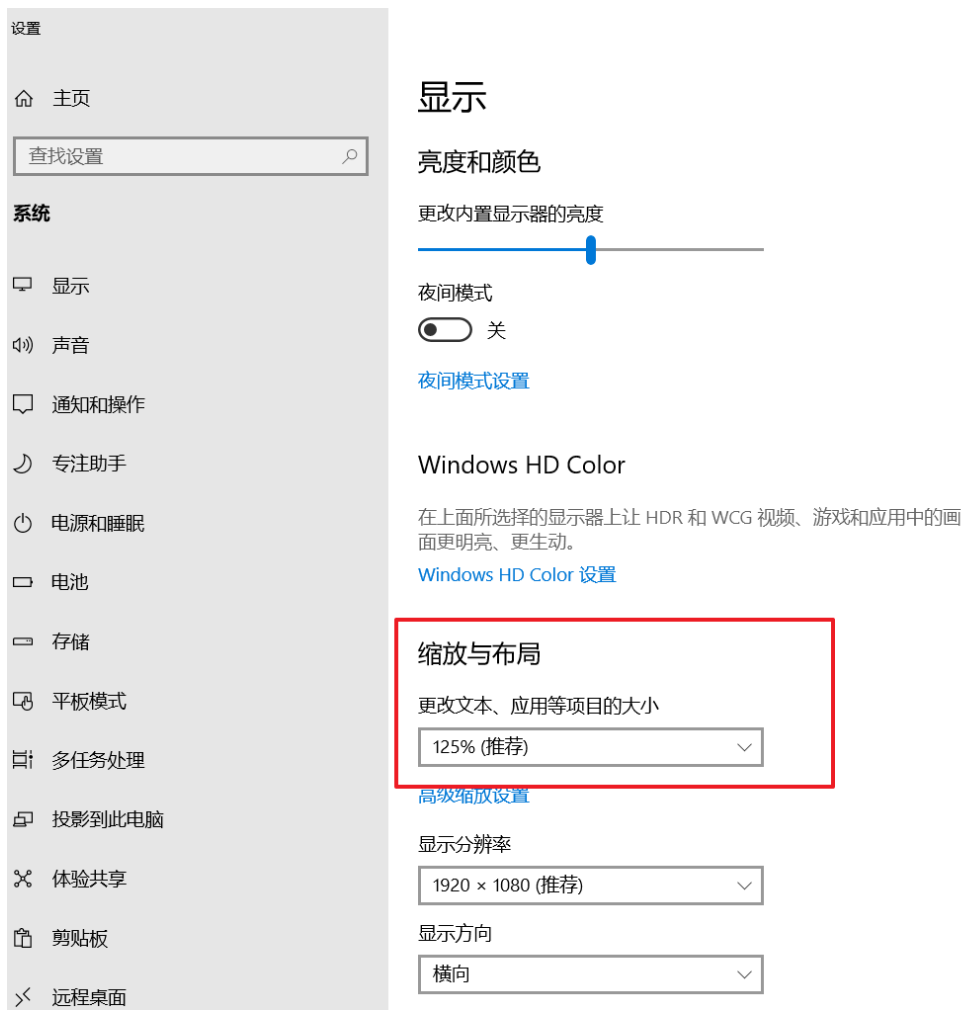
# 问题

大家在作业的时候,不要左边一个老师的源代码,右边一个自己的代码,这个效果不好,很差



1. 用markman量效果图,效果图怎么来的? 截图或者网页另存为图片

如果你使用的浏览器是谷歌浏览器,默认电脑的缩放与布局是125%比例,那么直接使用截图工具在谷歌浏览器中截图会被放大



解决方法:可以把缩放与布局换回100%以后截图或者使用国产浏览器360,搜狗浏览器等



直接使用截图工具截效果图

或者



如果有图片可以看图片属性,得到宽度高度;如果不是图片,就需要使用markman测试

2. `:hover`只能对**标签选择器**使用吗? 不是,是**所有的选择器**都可以使用

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
```

```
5 <title>Document</title>
6 <style>
7   div{
8     width: 100px;
9     height: 100px;
10    background-color:red;
11  }
12  div:hover{
13    background-color: blue;
14  }
15
16  p{
17    width: 200px;
18    height: 200px;
19    background-color: orange;
20  }
21  /*hover选择器可以给任何CSS选择器加*/
22  /*鼠标移上.p1以后的样式*/
23  .p1:hover{
24    background-color: green;
25  }
26
27  .father{
28    width: 300px;
29    height: 300px;
30    background-color: skyblue;
31  }
32  .father .son{
33    background-color: pink;
34  }
35
36  /*鼠标移上.father以后,.father里面的.son对应的样式*/
37  .father:hover .son{
38    background-color: red;
39  }
40 </style>
41 </head>
42 <body>
43   <div>div的内容</div>
44   <p class="p1">p的内容</p>
```

```
45 <h1 class="father">
46 父元素
47 <span class="son">子元素</span>
48 </h1>
49 </body>
50 </html>
```

#### 4. 遇到问题自己如何解决?

自己先尝试解决,比如使用浏览器检查工具查看

## 学习目标

- 能够使用背景简写属性
- 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

## 学习目标

- 能够使用背景简写属性 **background这个属性可以同时设置背景颜色,背景图片,背景平铺,背景定位**
- 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

盒子模型 = 内容 + 边框 + 内填充 + 外边距  
内容 width,height  
边框 border  
圆角边框 border-radius

内填充 padding,padding会撑开盒子的  
真实宽度高度,如何解决?

外边距 margin  
可以使用margin让盒子水平居中

## 背景简写(background)

background属性的值的**书写顺序**官方并没有强制标准的。为了可读性,建议大家如下写:

```
1 background: 背景颜色 背景图片 背景平铺 背景滚动 背景位置
2 background: red url(image.jpg) repeat-y scroll 50% 0 ;
```

## 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*背景简写(background)可以同时设置背景颜色,背景图片,背景平铺,背景定位,属性取值
      之间使用"空格"隔开,没有顺序之分,可以省略某个或者某几个*/
8     .div1{
9       width: 500px;
10      height: 500px;
11      border:2px solid red;
12      /* background-color:skyblue;
13      background-image:url(images/3.jpg);
14      background-repeat: no-repeat;
15      background-position: right bottom; */
16
17      background:url(images/3.jpg) skyblue no-repeat right bottom;
18    }
19    .div2{
20      /*使用背景简写属性有一个地方需要注意的*/
21      width: 500px;
22      height: 500px;
23      border:2px solid blue;
24
25      /*注意点1:大家不要同时在一个选择器里面写多次background这个简写属性*/
26      /*background:url(images/1.jpg) no-repeat;
27      background:right bottom;*/
28
29      /*注意点2:因为background会一次性设置所有背景相关属性,会先清空之前的属性值*/
30      /*background-image:url(images/1.jpg);*/
31      /*以下代码只是设置不重复,不设置其他背景相关样式*/
32      /*background: no-repeat;*/
33
34      /*重点:建议大家先使用背景简写,再使用单独背景样式*/
35      background:url(images/1.jpg);
36      background-repeat: no-repeat;
37      background-position: right bottom;
38    }
```

```

39  /*不要在单独背景样式中设置多个属性值*/
40  /*background-image:url(images/1.jpg) no-repeat;*/
41  }
42  </style>
43  </head>
44  <body>
45    <div class="div1">div1的内容</div>
46    <br/>
47    <div class="div2">div2的内容</div>
48  </body>
49  </html>

```

## 背景属性总结

属性名	属性取值	功能
background-color	英文颜色单词,十六进制,rgb,rgba	设置背景颜色
background-image	url(图片所在路径) 或 none无背景图 (默认的)	设置背景图片
background-repeat	repeat 或 no-repeat 或 repeat-x 或 repeat-y	设置背景平铺
background-position	长度值或百分比或方向名词	设置背景定位
background	可以同时设置背景颜色 背景图片 背景平铺 背景位置 没有书写顺序	背景属性简写,可以同时设置背景颜色 背景图片 背景平铺 背景位置 方便使用

## 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？

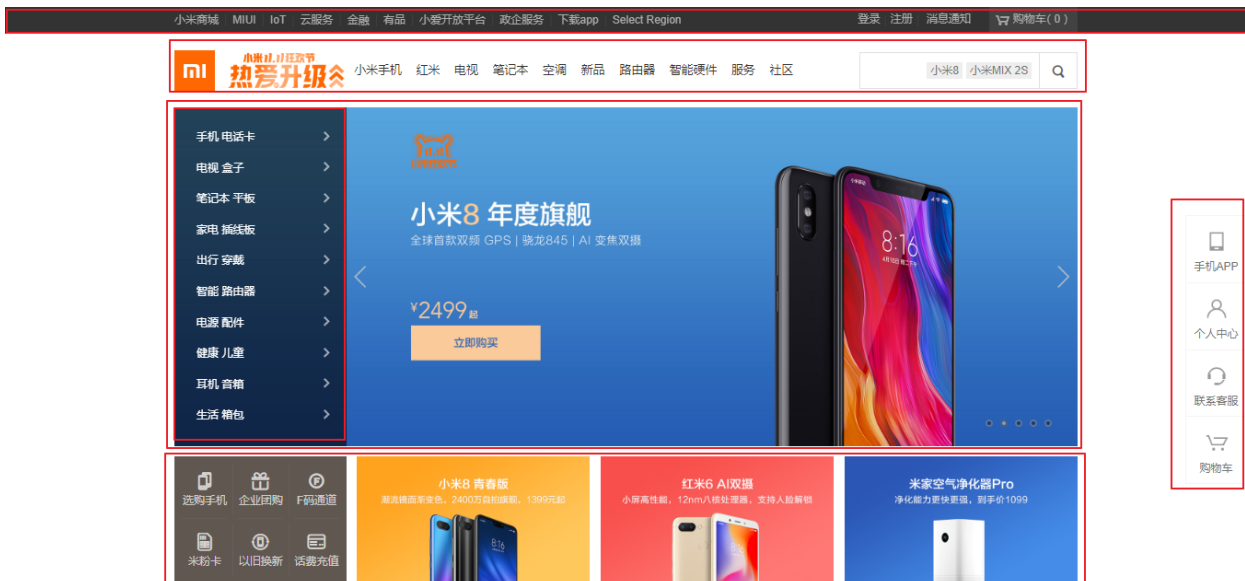


我们说过，行内元素比如 文字 类似牛奶，也需要一个盒子把他们装起来，我们前面学过的双标签都是一个盒子。有了盒子，我们就可以随意的，自由的，摆放位置了。



看透网页布局的本质:

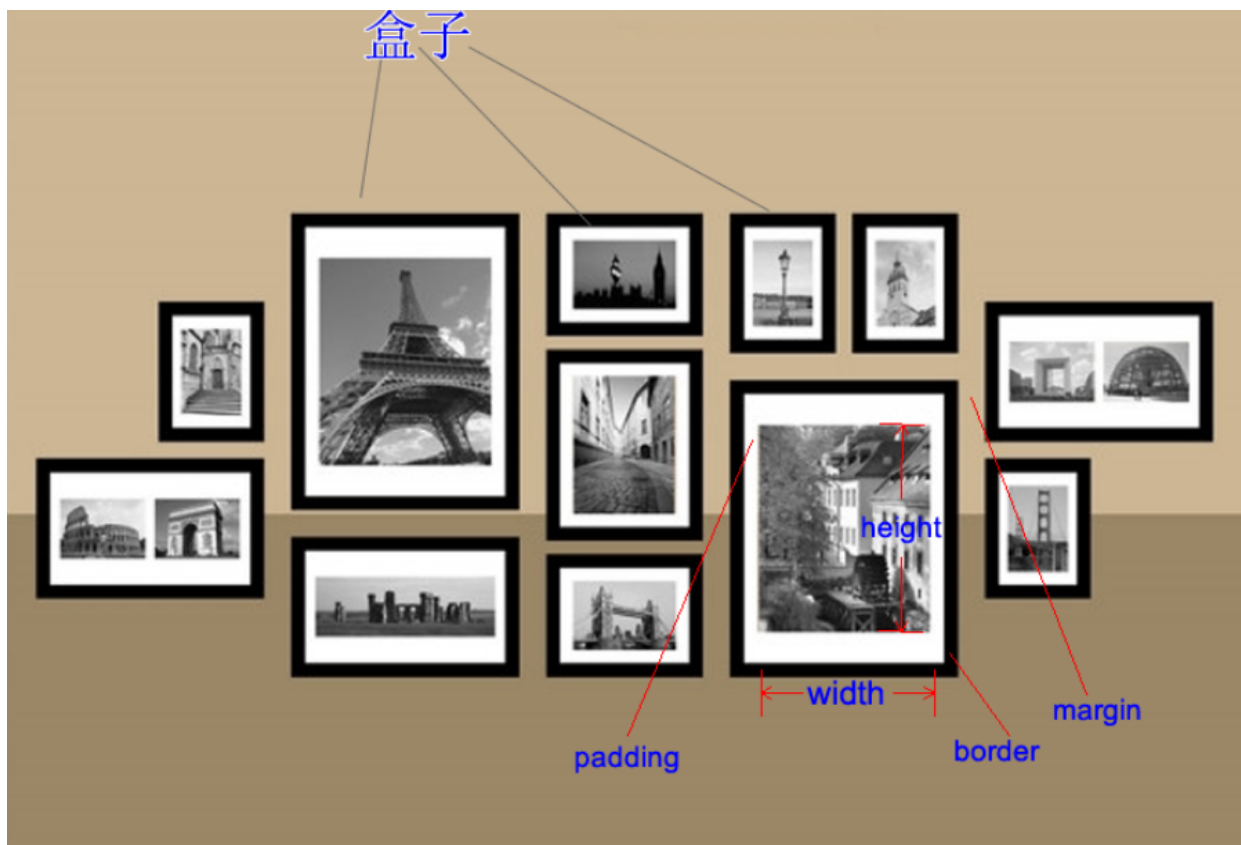
- 把网页元素比如文字图片等等，放入盒子里面。
- 然后利用CSS摆放盒子的过程，就是网页布局。
- 摆盒子的过程 就是网页布局



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:

```
1  css选择器{
2    border : border-width border-style border-color
3 }
```

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)
<b>solid(常用)</b>	<b>边框为单实线(最为常用的)</b>
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:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     .div1{
8       width: 300px;
9       height: 300px;
10      background: pink;
11
12
13      /*border设置的上下左右4个方向的边框*/
14
15      /*
16       border:border-width border-style border-color;
17       border:边框粗细 边框类型 边框颜色;
18
19       边框粗细取值是数值,带px单位,正值或者0,0代表无边框
20
21       边框类型取值 solid实线 dashed虚线 dotted点状线 none没有线 double双实线
22       其中solid跟none以及dashed取值用的最多
```

```
23
24 边框颜色取值 颜色值
25 */
26 border:5px solid red;
27 /*border:5px dashed red;*/
28 /*border:5px dotted red;*/
29 /*border:5px double red;*/
30 }
31
32 .div2{
33 width: 300px;
34 height: 300px;
35 background: skyblue;
36 /*可以单独设置某个方向的边框*/
37
38 /*上边框*/
39 /*border-top:边框粗细 边框类型 边框颜色;*/
40 border-top:5px solid green;
41
42 /*下边框*/
43 /*border-bottom:边框粗细 边框类型 边框颜色;*/
44 border-bottom:5px dashed orange;
45
46 /*左边框*/
47 /*border-left:边框粗细 边框类型 边框颜色;*/
48 border-left:5px dotted purple;
49
50 /*右边框*/
51 /*border-right:边框粗细 边框类型 边框颜色;*/
52 border-right:5px double red;
53 }
54
55 .div3{
56 width: 300px;
57 height: 300px;
58 background: orange;
59 /*我们平时使用边框的时候,可以先设置4个方向的边框,再单独设置某个或者某几个方向的边框*/
60 border:5px solid blue;
61
62 /*不要上边框线*/
```

```

63  /*border-top:5px none blue;*/
64  /*我们更喜欢直接使用一个none取值,去掉边框线*/
65  border-top:none;
66
67  border-right:10px dashed red;
68  }
69  </style>
70  </head>
71  <body>
72  <div class="div1">div1的内容</div>
73
74  <br/>
75
76  <div class="div2">div2的内容</div>
77
78  <br/>
79
80  <div class="div3">div3的内容</div>
81  </body>
82  </html>

```



## 盒子边框写法总结表

设置内容	样式属性	常用属性值
上边框	border-top-style:样式; border-top-width:宽	



	度;border-top-color:颜色;border-top:宽度 样式 颜色;	
下边框	border-bottom-style:样式;border-bottom-width:宽度;border-bottom-color:颜色;border-bottom:宽度 样式 颜色;	
左边框	border-left-style:样式; border-left-width:宽度;border-left-color:颜色;border-left:宽度 样式 颜色;	
右边框	border-right-style:样式;border-right-width:宽度;border-right-color:颜色;border-right:宽度 样式 颜色;	
样式综合设置	border-style:上边 [右边 下边 左边];	none无(默认)、solid单实线、dashed虚线、dotted点线、double双实线
宽度综合设置	border-width:上边 [右边 下边 左边];	像素值
颜色综合设置	border-color:上边 [右边 下边 左边];	颜色值、#十六进制、rgb(r,g,b)、rgb(r%,g%,b%)
边框综合设置	border:四边宽度 四边样式 四边颜色;	

## 圆角边框(CSS3)

语法格式:

```

1  css选择器{
2      border-radius:长度值或百分比;
3  }
```

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:

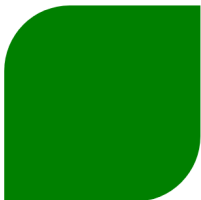
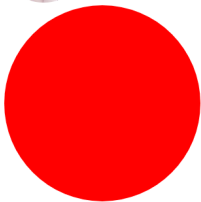
```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <style>
7          div{
8              width: 300px;
```



```
9 height: 300px;
10 }
11 .div1{
12 background: pink;
13 text-align: center;
14 line-height: 300px;
15 border:5px solid blue;
16 /*border-radius设置圆角边框属性,取值数值带px,或者百分比*/
17 /*数值取值只能是正值,不能是负值*/
18 border-radius: 50px;
19 }
20 .div2{
21 background: skyblue;
22 line-height: 300px;
23 text-align: center;
24 /*圆角边框可以单独使用*/
25 border-radius: 50px;
26 }
27
28 img{
29 /*圆角边框属性可以给元素设置,也可以给图片设置*/
30 border-radius: 100px;
31 }
32 .div3{
33 background: red;
34 /*border-radius: 10%;*/
35
36 /*圆角边框取值"50%"的时候,可以让一个"正方形"变成一个"圆形"*/
37 border-radius: 50%;
38 }
39 p{
40 width: 200px;
41 height: 50px;
42 background: blue;
43 /*椭圆*/
44 border-radius: 50%;
45 }
46
47 .div4{
48 background: green;
```

```
49  /*border-radius取值可以写一个值也可以写多个值,多个值老师只给大家讲一个4个取值
    的*/
50
51  /*border-radius:左上角 右上角 右下角 左下角; 记忆方法:顺时针 */
52  border-radius: 100px 0px 100px 0px;
53  }
54  .img2{
55  border-radius: 100px 0px 100px 0px;
56  }
57  </style>
58  </head>
59  <body>
60  <div class="div1">div1的内容</div>
61  <br/>
62  <div class="div2">div2的内容</div>
63  <br/>
64  
65  <br/>
66  <div class="div3"></div>
67  <br/>
68  <p></p>
69  <br/>
70  <div class="div4"></div>
71  <br/>
72  
73  </body>
74  </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!

语法:

```
1 table{ border-collapse:collapse; }  
2 说明:  
3 collapse 单词是合并的意思  
4 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>
1	鬼吹灯	1	356	3560	<a href="#">贴吧</a> <a href="#">图片</a> <a href="#">百科</a>

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     table{
8       /*CSS给table标签设置边框,只有最外层有*/
9       border:1px solid #ff3aa4;
10
11       /*我们以前是通过html属性,rules="all"合并边框线*/
12
13       /*现在我们可以使用css的一个属性 border-collapse:collapse;*/
14       /*注意: border-collapse:collapse只能给table标签设置*/
15       border-collapse: collapse;
16     }
17     /*CSS给tr标签设置边框无效*/
18     /*tr{
19       border:1px solid #ff3aa4;
20     }*/
21
22
23     td,th{
24       border:1px solid #ff3aa4;
```

```
25  }
26
27  table tr:nth-child(odd){
28  background: #ffc0cb;
29  }
30  table tr th{
31  background: #ff69b4;
32  }
33  </style>
34  </head>
35  <body>
36  <table>
37  <caption><h3>小说排行榜</h3></caption>
38  <tr>
39  <th>排名</th>
40  <th>关键词</th>
41  <th>趋势</th>
42  <th>今日搜索</th>
43  <th>最近七日</th>
44  <th>相关链接</th>
45  </tr>
46  <tr>
47  <td>1</td>
48  <td>鬼吹灯</td>
49  <td>1</td>
50  <td>356</td>
51  <td>3560</td>
52  <td>
53  <a href="#">贴吧</a>
54  <a href="#">图片</a>
55  <a href="#">百科</a>
56  </td>
57  </tr>
58  <tr>
59  <td>1</td>
60  <td>鬼吹灯</td>
61  <td>1</td>
62  <td>356</td>
63  <td>3560</td>
64  <td>
```

```
65 <a href="#">贴吧</a>
66 <a href="#">图片</a>
67 <a href="#">百科</a>
68 </td>
69 </tr>
70 <tr>
71 <td>1</td>
72 <td>鬼吹灯</td>
73 <td>1</td>
74 <td>356</td>
75 <td>3560</td>
76 <td>
77 <a href="#">贴吧</a>
78 <a href="#">图片</a>
79 <a href="#">百科</a>
80 </td>
81 </tr>
82 <tr>
83 <td>1</td>
84 <td>鬼吹灯</td>
85 <td>1</td>
86 <td>356</td>
87 <td>3560</td>
88 <td>
89 <a href="#">贴吧</a>
90 <a href="#">图片</a>
91 <a href="#">百科</a>
92 </td>
93 </tr>
94 <tr>
95 <td>1</td>
96 <td>鬼吹灯</td>
97 <td>1</td>
98 <td>356</td>
99 <td>3560</td>
100 <td>
101 <a href="#">贴吧</a>
102 <a href="#">图片</a>
103 <a href="#">百科</a>
104 </td>
```

```
105 </tr>
106 <tr>
107 <td>1</td>
108 <td>鬼吹灯</td>
109 <td>1</td>
110 <td>356</td>
111 <td>3560</td>
112 <td>
113 <a href="#">贴吧</a>
114 <a href="#">图片</a>
115 <a href="#">百科</a>
116 </td>
117 </tr>
118 <tr>
119 <td>1</td>
120 <td>鬼吹灯</td>
121 <td>1</td>
122 <td>356</td>
123 <td>3560</td>
124 <td>
125 <a href="#">贴吧</a>
126 <a href="#">图片</a>
127 <a href="#">百科</a>
128 </td>
129 </tr>
130 </table>
131 </body>
132 </html>
```

## 轮廓属性outline

是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

```
1 outline : outline-color ||outline-style || outline-width
```

但是我們都不关心可以设置多少，我们平时都是去掉的。

最直接的写法是：`outline: 0;` 或者 `outline: none;`

**举例:**

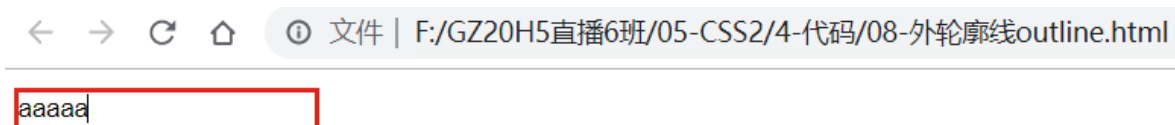
```
1 <!DOCTYPE html>
```

```

2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     input{
8       border:2px solid #e2231a;
9       /*外轮廓线outline,一般我们使用的是outline:none,去掉外轮廓线*/
10      outline:none;
11    }
12  </style>
13 </head>
14 <body>
15   <form action="index.html" method="get">
16     <input type="text"/>
17   </form>
18 </body>
19 </html>

```

### 效果图:



### 练习:

用户名:

邮 箱:

搜索一下:

说明: **E:focus**代表E元素获取焦点以后的样式



用户名:

邮 箱:

搜索一下:

获取焦点以后

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     input{
8       /*去掉外轮廓线*/
9       outline:none;
10    }
11    .username{
12      border:1px dashed gray;
13      background: #e7eefe;
14    }
15    /*文本框获取鼠标焦点对应的样式*/
16    .username:focus{
17      border:1px dashed #ffc8d2;
18      background: #fff7fb;
19    }
20    .email{
21      /*去掉所有方向的边框*/
22      border:none;
23      border-bottom:1px dashed red;
24    }
25  </style>
26 </head>
27 <body>
28   <form action="index.php" method="get">
29     <p>用户名: <input type="text" class="username"/></p>
30
31     <p>邮 箱: <input type="text" class="email"/></p>
32
```

```
33 <p>搜索一下: <input type="text" class="search" placeholder="请输入..." />
</p>
34 </form>
35 </body>
36 </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:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 .div1{
```

```

8  width: 200px;
9  height: 200px;
10 background: pink;
11 border:2px solid blue;
12 /*padding指的是内容到边框的距离*/
13 /*padding取值是数值带px单位,"只能写正值,负值无效"*/
14 /*padding-top上填充*/
15 padding-top:50px;
16
17 /*padding-left左填充*/
18 padding-left: 10px;
19 }
20
21 .div2{
22 width: 200px;
23 border:2px solid red;
24 background: skyblue;
25 /*padding-bottom下填充*/
26 padding-bottom: 20px;
27
28 /*padding-right右填充*/
29 text-align:right;
30 padding-right: 20px;
31 }
32 </style>
33 </head>
34 <body>
35 <div class="div1">div1的内容</div>
36
37 <br/>
38
39 <div class="div2">div2的内容</div>
40 </body>
41 </html>

```

#### 举例:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>

```

```
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7   div{
8     width: 200px;
9     height: 200px;
10  }
11  .one{
12    background: pink;
13    /*padding:一个值代表上下左右都是10px*/
14    padding:10px;
15  }
16
17  .two{
18    background: skyblue;
19    /*padding:两个值;第一个值代表上下,第二个值代表左右*/
20    padding:10px 20px;
21  }
22
23  .three{
24    background: lightgreen;
25    /*padding:三个值;第一个值代表上,第二个值代表左右,第三个值代表下*/
26    padding:10px 20px 30px;
27  }
28
29  .four{
30    background: deeppink;
31    /*padding:四个值;第一个值代表上,第二个值代表右,第三个值代表下,第四个代表左;顺
    时针*/
32    padding:10px 20px 30px 40px;
33  }
34
35
36
37 </style>
38 </head>
39 <body>
40 <!-- .one+.two+.three+.four -->
41 <div class="one">one</div>
42 <br/>
43
```

```
44 <div class="two">two</div>
45 <br/>
46
47 <div class="three">three</div>
48 <br/>
49
50 <div class="four">four</div>
51 </body>
52 </html>
```

## 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

- 1 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
- 2 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:**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 .one{
8 width: 200px;
9 height: 200px;
10 background: pink;
11 /*padding会撑开盒子真实宽度跟高度*/
12 /*左填充*/
13 padding-left:20px;
14 /*右填充*/
```

```

15 padding-right:30px;
16 /*上填充*/
17 padding-top:40px;
18 /*下填充*/
19 padding-bottom:100px;
20 /*四个方向的边框*/
21 border:3px solid blue;
22
23 /*算真实宽度高度很重要,因为我们明天要学习浮动,需要用到这个知识*/
24 /*结论*/
25 /*盒子的真实宽度 = 宽度 + 左填充 + 右填充 + 左边框 + 右边框*/
26 /*盒子的真实高度 = 高度 + 上填充 + 下填充 + 上边框 + 下边框*/
27
28 /*验证一下*/
29 /*one盒子的真实宽度 = 200 + 20 + 30 + 3 + 3 = 256*/
30 /*one盒子的真实高度 = 200 + 40 + 100 + 3 + 3 = 346*/
31 }
32 </style>
33 </head>
34 <body>
35 <div class="one">one</div>
36 </body>
37 </html>

```

**保证盒子真实宽度高度不变: 增加了多少的内填充就要减去盒子里面内容的宽度和高度**

**举例:**

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 .one{
8 /*width: 200px;*/
9 /*width: 100px;*/
10 width: 80px;
11
12 /*height: 200px;*/

```

```

13  /*height: 150px;*/
14  height: 140px;
15
16  background: skyblue;
17  /*左填充*/
18  padding-left: 100px;
19  /*右填充*/
20  padding-right: 20px;
21  /*上填充*/
22  padding-top: 50px;
23  /*下填充*/
24  padding-bottom: 10px;
25
26  /*如何保证盒子真实宽度高度不变?*/
27  /*一句话:多了多少的填充,就减去多少相应宽度或者高度*/
28  }
29  </style>
30 </head>
31 <body>
32   <div class="one">one</div>
33 </body>
34 </html>

```

## 练习:

- 1 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，
- 2 请问盒子的width和height应该设置成大？

## 总结:

- 1 border和padding会影响盒子的大小，margin不会
- 2 盒子的真实宽度：width+左右border+左右padding
- 3 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，由于默认块级元素的宽度是父元素的百分比,所以给这个盒子指定左右padding，则不会撑开盒子。

## 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*padding不影响块元素盒子宽度情况*/
8     .one{
9       /*不给块元素设置宽度,默认"块元素的宽度是父元素的100%"/
10      background-color: pink;
11      /*设置左填充*/
12      padding-left: 100px;
13      /*设置右填充*/
14      padding-right: 50px;
15
16      /*小结:如果一个块级元素没有设置宽度,那么设置左右填充的时候,就不需要减去对应的
17      左右填充距离*/
18    }
19  </style>
20 </head>
21 <body>
22   <div class="one">one</div>
23 </body>
24 </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**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*给元素设置背景图片与背景颜色是不受padding影响的,"padding只影响元素里面的内
8     容"*/
9     .one{
10      width: 500px;
11      height: 500px;
```



```
11 border:2px solid blue;
12 /*background: pink;*/
13 background:url(images/1.jpg);
14 padding-left: 10px;
15 padding-top: 10px;
16 }
17 </style>
18 </head>
19 <body>
20 <div class="one">one的内容</div>
21 </body>
22 </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()

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style type="text/css">
7   div {
8     width: 200px;
9     height: 200px;
10    border: 1px solid #000000;
11    border-top: 5px solid blue;
```

```

12 padding: 50px;
13 padding-left: 100px;
14 }
15 </style>
16 </head>
17 <body>
18 <div></div>
19 </body>
20 </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

## 练习

要求使用 **ul li a** 标签完成，并且使用 **背景图片** 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



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 a{
8 text-decoration: none;
9 }

```

```

10  a:hover{
11  text-decoration: underline;
12  color:pink;
13  }
14  li{
15  /*去掉列表项目符号*/
16  list-style:none;
17  }
18  ul li{
19  background:url(images/li.gif) no-repeat left center;
20
21  /*想看一个元素的位置在哪里,可以给这个元素设置边框或者设置背景颜色*/
22  /*border:1px solid red;*/
23
24  /*text-indent:1.2em;*/
25  padding-left: 25px;
26  }
27  </style>
28  </head>
29  <body>
30  <ul>
31  <li><a href="#">大明星:姜潮魔性拜年道晚安</a></li>
32  <li><a href="#">软萌正太徐浩演绎《小幸运》</a></li>
33  <li><a href="#">漫威绝逼好看的电影镜头合集</a></li>
34  <li><a href="#">从没见过这么搞笑的祖孙组合</a></li>
35  <li><a href="#">史上最容易挨揍的自助餐吃法</a></li>
36  </ul>
37  </body>
38  </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:**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*外边距(margin)的基本用法跟padding用法基本一样*/
8     /*margin指的是盒子到盒子之间的距离,可以理解为盒子外面的距离*/
9     /*margin取值可以是数值带px,正值负值都可以,还有一个特殊的值auto自动的意思,待会
    会讲auto*/
10
11     /*margin-top上外边距*/
12     /*margin-bottom下外边距*/
13     /*margin-left左外边距*/
14     /*margin-right右外边距*/
15
16     div{
17       width: 200px;
18       height: 200px;
19     }
20     .one{
21       background: red;
22       margin-top: 100px;
23       margin-bottom: 50px;
24     }
25     .two{
26       background: blue;
27       margin-left: 100px;
28     }
29     .three{
30       background: green;
31       /*先使用浮动属性,这个浮动属性明天我们会讲*/
32       float: right;
33       margin-right: 100px;
34     }
35   </style>
```

```
36 </head>
37 <body>
38 <div class="one">one</div>
39 <div class="two">two</div>
40 <div class="three">three</div>
41 </body>
42 </html>
```

#### 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 /*margin的取值也可以有一个值,两个值,三个值,四个值,用法跟padding一致*/
8 div{
9 width: 150px;
10 height: 150px;
11 }
12 .one{
13 background: red;
14 margin:10px;
15 }
16 .two{
17 background: green;
18 margin:10px 20px;
19 }
20 .three{
21 background: blue;
22 margin:10px 20px 30px;
23 }
24 .four{
25 background: pink;
26 margin:10px 20px 30px 40px;
27 }
28 </style>
29 </head>
30 <body>
31 <div class="one">one</div>
```

```
32 <div class="two">two</div>
33 <div class="three">three</div>
34 <div class="four">four</div>
35 </body>
36 </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：

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

### 举例：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 .box{
8 width: 1000px;
9 height:200px;
10 background: lightgreen;
11 /*可以让一个盒子实现水平居中，需要满足一下三个条件:*/
12 /*1.必须是块级元素。*/
13 /*2.盒子必须指定了小于父元素100%的宽度(width)*/
14 /*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。*/
15 /*auto代表自动分配空间*/
16
17 /*把空白空间自动分配给左边*/
18 /*margin-left:auto;*/
19
20 /*把空白空间自动分配给右边*/
21 /*margin-right:auto;*/
22
23 /*同时设置左右margin的取值为auto*/
```

```
24 margin-left:auto;
25 margin-right:auto;
26 }
27 </style>
28 </head>
29 <body>
30 <div class="box">box盒子的内容</div>
31 </body>
32 </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 设置左右margin 改为 auto
- 3.图片水平居中
- 4.背景图片水平居中

### 举例:

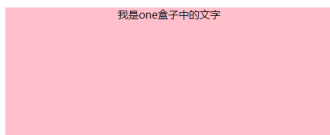
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 .one{
8 width: 500px;
9 height: 200px;
10 background: pink;
11 text-align:center;
12 }
13
14 .two{
15 /*三个条件*/
16 /*1.块元素*/
17 /*2.设置一个小于100%的宽度*/
18 width:500px;
19 height: 200px;
20 background: skyblue;
21 /*3.设置左右外边距为auto*/
```

```
22 margin-left:auto;
23 margin-right: auto;
24
25 /*margin:0 auto;*/
26
27 /*上下外边距设置auto无效*/
28 /*margin:auto;*/
29 }
30 .three{
31 width: 500px;
32 height: 300px;
33 background: lightgreen;
34 text-align:center;
35 }
36
37 .four{
38 /*width: 500px;*/
39 width:310px;
40 height: 300px;
41 background: deeppink;
42 padding-left: 190px;
43 }
44
45 .five{
46 width: 500px;
47 height: 300px;
48 background: orange;
49 }
50 .five img{
51 margin-left: 190px;
52 }
53
54 .six{
55 width: 500px;
56 height: 300px;
57 background: purple url(images/1.jpg) no-repeat center top;
58 }
59 </style>
60 </head>
61 <body>
```



```
62 <h2>文字水平居中 text-align:center</h2>
63 <div class="one">我是one盒子中的文字</div>
64
65 <h2>盒子水平居中 使用margin</h2>
66 <div class="two">two盒子</div>
67
68 <h2>图片在盒子里面水平居中 text-align:center;因为img标签是行内块元素,text-a
    lign可以生效</h2>
69 <div class="three">
70 
71 </div>
72
73 <h2>图片在盒子里面水平居中 使用父元素的padding-left</h2>
74 <div class="four">
75 
76 </div>
77
78 <h2>图片在盒子里面水平居中 使用img标签的左外边距</h2>
79 <div class="five">
80 
81 </div>
82
83 <h2>背景图片水平居中 使用背景定位属性</h2>
84 <div class="six"></div>
85 </body>
86 </html>
```

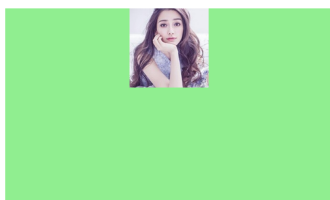
文字水平居中 text-align:center



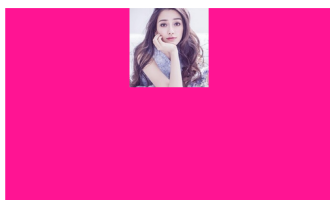
盒子水平居中 使用margin



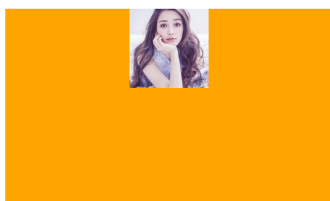
图片在盒子里面水平居中 text-align:center;因为img标签是行内块元素,text-align可以生效



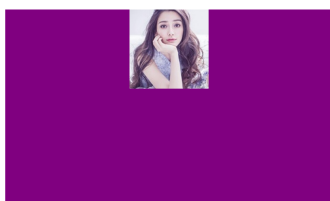
图片在盒子里面水平居中 使用父元素的padding-left



图片在盒子里面水平居中 使用img标签的左外边距



背景图片水平居中 使用背景定位属性



## 清除元素的默认内外边距 重点

为了更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：

```
1 * {  
2   padding:0;          /* 清除内边距 */  
3   margin:0;           /* 清除外边距 */  
4 }
```

也可以这样做

```
1 html,body,h1,h2,h3,h4,h5,h6,div,span,p,ul,li,dl,dt,dd,img,input,ol{  
2   margin: 0;
```

```
3 padding:0;
4 }
```

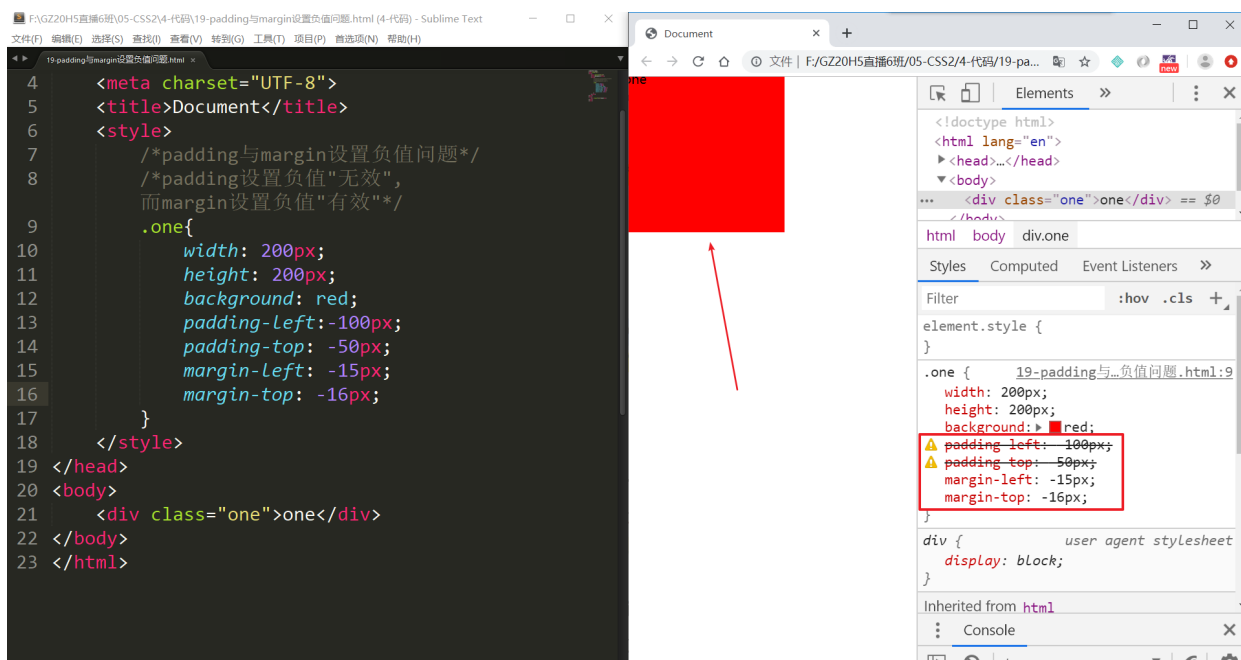
## 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*因为默认有些标签自带margin或者padding,这样对我们后续布局非常不利*/
8     /*所以我们喜欢先把自带的margin和padding清0*/
9
10    /*我们会使用通配符选择器,选中页面所有,清除内填充与外边距*/
11    *{
12      margin: 0;
13      padding: 0;
14    }
15  </style>
16 </head>
17 <body>
18   <h1 style="background:skyblue">标题1</h1>
19   <h2>标题2</h2>
20   <h3>标题3</h3>
21   <h4>标题4</h4>
22   <h5>标题5</h5>
23   <h6>标题6</h6>
24   <p>pppppp</p>
25   <ul>
26     <li>张三</li>
27     <li>李四</li>
28     <li>王五</li>
29   </ul>
30 </body>
31 </html>
```

## padding与margin设置负值问题

padding设置负值无效,margin设置负值有效

举例:

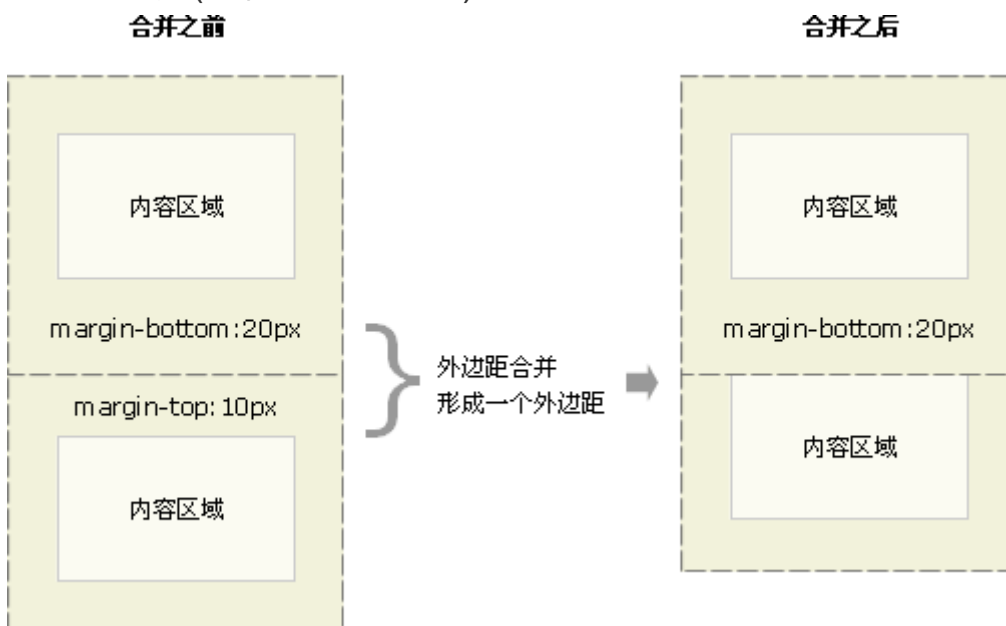


## margin外边距合并现象(margin常见问题) 扩展知识

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)。



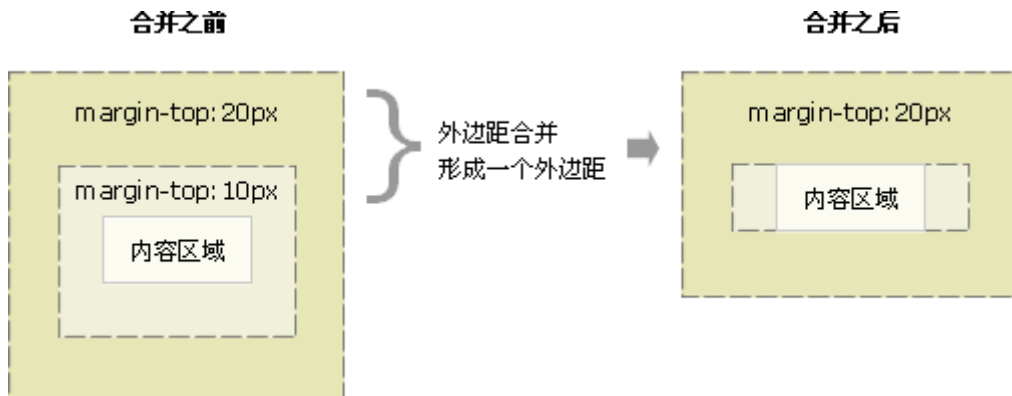
解决方案:如果是垂直方向的盒子在设置外边距时 只需要设置一个就好

## 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    div{
12      width: 200px;
13      height: 200px;
14    }
15    .one{
16      background: red;
17      /*上面的盒子添加下外边距*/
18      /*margin-bottom: 20px;*/
19
20      /*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
21      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
22      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)。
23      */
24
25      /*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*/
26      margin-bottom: 30px;
27    }
28    .two{
29      background: blue;
30      /*下面的盒子添加上外边距*/
31      /*margin-top: 10px;*/
32    }
33  </style>
34 </head>
35 <body>
36   <div class="one">one</div>
37   <div class="two">two</div>
38 </body>
39 </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填充。
2. 可以为父元素添加overflow:hidden。

### 举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*嵌套(父子关系)块元素垂直外边距的合并*/
8     /*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
9      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
10     为0，也会发生合并。*/
11     *{
12       margin: 0;
13       padding: 0;
14     }
15     .father{
16       width: 500px;
17       height: 500px;
18       background: red;
19       /*给父盒子添加上外边距*/
20       margin-top: 20px;
```

```

21  /*可以为父元素定义1像素的上边框或1像素上内填充。*/
22  /*border-top:1px solid orange;*/
23  /*padding-top: 1px;*/
24
25  /*可以为父元素添加overflow:hidden。 推荐使用 overflow:hidden代表溢出隐藏*/
26  overflow:hidden;
27  }
28  .father .son{
29  width: 200px;
30  height: 200px;
31  background: blue;
32  /*给子盒子添加上外边距*/
33  margin-top: 10px;
34  }
35  </style>
36  </head>
37  <body>
38  <div class="father">
39  <div class="son"></div>
40  </div>
41  </body>
42  </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**

**解决方案:**

1. 可以为父元素定义1像素的上边框或上内填充
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:**

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <title>Document</title>
6  <style>
7  /*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*/

```

```
8  *{
9  margin: 0;
10 padding: 0;
11 }
12 .father{
13 width: 500px;
14 height: 500px;
15 /*height: 400px;*/
16
17 background: orange;
18
19 /*解决方案:*/
20 /*可以为父元素定义1像素的上边框或上内填充*/
21 /*border-top:1px solid blue;*/
22 /*padding-top: 1px;*/
23
24 /*可以为父元素添加overflow:hidden 推荐使用*/
25 overflow:hidden;
26
27 /*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*/
28 /*padding-top: 100px;*/
29 }
30 .father .son{
31 width: 200px;
32 height: 200px;
33 background: red;
34 margin-top: 100px;
35 }
36 </style>
37 </head>
38 <body>
39 <div class="father">
40 <div class="son"></div>
41 </div>
42 </body>
43 </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)。

```
1 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。**

## 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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
```

```
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 /*第二步:书写CSS初始化样式*/
8 /*初始化样式开始,注意:不要在初始化样式中添加一些无关的代码,特别的代码,可以使用
  后代选择器书写*/
9 /*清除所有标签的内外边距*/
10 *{
11     margin: 0;
12     padding: 0;
13 }
14 /*去掉列表标签项目符号*/
15 li{
16     list-style:none;
17 }
18 /*超链接的美化*/
19 a{
20     text-decoration: none;
21     color:#202020;
22 }
23 a:hover{
24     text-decoration: underline;
25 }
26 /*初始化样式结束*/
27
28 /*第三步:写具体样式,顺序从大到小,先整体后细节,从左到右,从上到下*/
29 body{
30     background: #eeeeee;
31 }
32
33 .box{
34     width: 372px;
35     height: 283px;
36     margin-top: 100px;
37     margin-left: auto;
38     margin-right: auto;
39     border:1px solid #cccccc;
40     padding-left: 19px;
41     padding-right: 19px;
42 }
```

```
43  .box h2{
44  height: 58px;
45  border-bottom: 1px solid #cccccc;
46  /*字体大小就是量文字高度即可*/
47  font-size:22px;
48  /*如果行高距离比高度大,单行文字就会垂直偏下*/
49  line-height: 78px;
50  }
51  .box h2 span{
52  font-size:18px;
53  }
54  .box ul {
55  /*看一个元素的位置可以给这个元素设置边框或者背景颜色*/
56  /*background: skyblue;*/
57  margin-top: 12px;
58  }
59  .box ul li{
60  height: 38px;
61  border-bottom: 1px dashed #cccccc;
62  line-height: 38px;
63  text-indent:25px;
64
65  /*为什么不需要减宽度*/
66  /*padding-left: 25px;*/
67  }
68  </style>
69  </head>
70  <body>
71  <!-- 实现思路 -->
72  <!-- 第一步：根据效果图写HTML结构 -->
73
74  <!-- 外面最大的盒子 -->
75  <div class="box">
76  <!-- 上部分 -->
77  <h2>最新文章<span>/New Articles</span></h2>
78
79  <!-- 下部分 -->
80  <ul>
81  <li><a href="#">北京招聘网页设计，平面设计，php</a></li>
82  <li><a href="#">体验javascript的魅力</a></li>
```

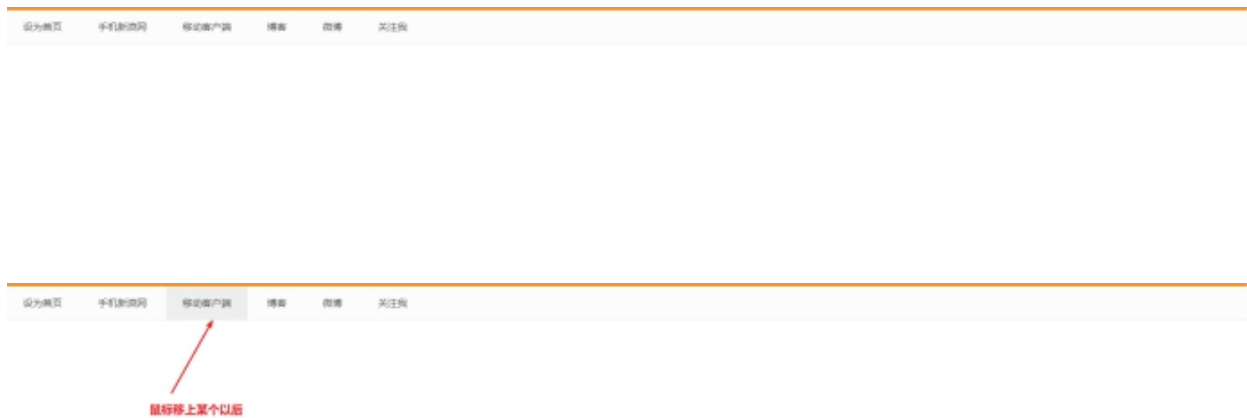
```
83 <li><a href="#">jquery世界来临</a></li>
84 <li><a href="#">网页设计师的梦想</a></li>
85 <li><a href="#">jquery中的链式编程是什么</a></li>
86 </ul>
87
88 </div>
89 </body>
90 </html>
```

## 今日总结

## 今日作业

使用xmind总结当天知识点

## 新浪导航栏案例



## 爱宠知识

爱宠知识
养狗比养猫对健康更利
日本正家犬亮相，你怎么看柴犬
狗狗歌曲《新年旺旺》
带空调风，开车带空调需要注意什么？
【爆笑】这跑跑太不给力了
跑跑与男星相同着装拍有型造型照
跑跑首个阶段健康大事件
调皮宠物狗躲在沙发里的搞笑瞬间
为什么每次大小便后，会用脚踢土？

搜索趣图

搜索趣图

- GIF: 小胖墩游泳被卡 被救后一脸无辜
- GIF: 小胖墩游泳被卡 被救后一脸无辜
- GIF: 小胖墩游泳被卡 被救后一脸无辜

鲁能-本题选做

鲁能热身赛16-0大胜业余队 野牛4球蒙蒂略亮相

2015年08月02日 18:18 新浪体育 报道 共有20119.170人参与 收藏本文



鲁能热身赛16-0大胜

新浪体育讯 北京时间8月2日傍晚，山东鲁能俱乐部在训练基地与业余球队山东太阳足球俱乐部进行了一场热身赛，最终鲁能以16-0大胜对手，阿洛伊西奥上演大四喜，韩鹏和蒙蒂略均取得进球，尤西雷梅开二度，埃尔克森、赵明剑、张立利、成源均取得进球。

**鲁能泰山足球新闻：**鲁能泰山4鲁能热身赛16-0大胜业余足球队，阿洛伊西奥大四喜，韩鹏进球，尤西雷梅开二度，埃尔克森、赵明剑、张立利、成源均取得进球。  
2015-08-02 17:54 来自鲁能足球报 原创(4) (编辑) (评论)

由于国足近期无杯比赛，中超联赛获得了两周的间歇期，山东鲁能训练赛期间，也邀请了一支业余球队进行热身赛。尽管王大雷、王永珀、杨旭、蒿俊闵等人随队参加东亚足协杯，但鲁能还是派出阵容最强阵容，以16-0大胜了对手，蒿俊闵上半场就取得了9-0的领先，鲁能与业余队比赛，胜者并不重要，这场比赛的大比分胜利也能看出鲁能队目前状态不错。

下半场中超第2轮即将打响，山东鲁能客场挑战北京国安，两支球队球队的交锋对决，无疑备受关注。由于国足近期无杯比赛，鲁能阵中的国脚们能够有更多时间来热身，而以蒿俊闵为首的鲁能阵容，实力也绝对不容小觑。(陈亮)

**文章关键词：**中超 鲁能 阿洛伊西奥 蒙蒂略

点击下载【[新浪体育客户端](#)】，赛事视频直播尽在掌握