

CSS页面布局

盒子模型：



内容区

内容区有三个属性：

- width height 控制内容区的高度与宽度
- overflow 属性控制当内容区的信息太多并超过内容区所占的范围时，对于溢出内容的处理方式

值	描述
visible	默认值，溢出内容不会被裁切，会呈现在元素框之外
hidden	溢出内容不可见
scroll	溢出内容被裁切，但是可以通过滚动条查看
auto	由浏览器自己决定

例子：

```
<style type="text/css">
  div{
    width:400px;
    height:200px;
    overflow:scroll; /*增加滚动条*/
  }
</style>
```

边框

边框就是围绕元素的内容和内边距的一条或多条线。属性如下：

值	描 述
border-top-width	设置元素的上边框的宽度
border-top-style	设置元素的上边框的样式
border-top-color	设置元素的上边框的颜色
border-top/bottom/left/right	边框的简写形式，用于把边框的所有属性设置到一个声明中；可以按如下顺序设置属性：border-top-width、border-top-style、border-top-color
border-width	边框宽度的简写形式，用于设置元素所有边框的宽度，或者单独地为各边框设置宽度
border-style	边框样式的简写形式，用于设置元素所有边框的样式，或者单独地为各边设置边框样式
border-color	设置元素的所有边框中可见部分的颜色，或者单独地为各边设置边框颜色
border	边框的简写形式，用于把针对四个边的属性设置在一个声明
border-radius	圆角边框的简写形式，用于设置四个border-*-radius属性，CSS 3新增
box-shadow	向框添加一个或多个阴影，CSS 3新增

注：元素的边框在元素的背景之上，当边框为点线边框或者虚线框时，元素的背景就会显示出来。

1.边框宽度

属性：**border-width**

边框宽度的参数可以是1~4个。可由数值或关键字确定。关键字有:thin medium thick

- 当有4个参数时，按照Top-->Right-->Bottom-->Left顺序赋值。（上右下左）
- 当有3个参数时，按照Top-->Right+Left-->Bottom顺序赋值。（左右一致）
- 当有2个参数时，按照Top+Bottom-->Right+Left顺序赋值。（上下、左右一致）
- 当有1个参数时，按照Top+Right+Bottom+Left顺序赋值。（全部一致）

2.边框样式

属性：**border-style**

参数赋值方式与边框宽度类似。

值：

值	描述
none	无边框
hidden	隐藏边框
dotted	定义点状边框，在大多数浏览器中呈现为实线
dashed	定义虚线，在大多数浏览器中呈现为实线
solid	定义实线
double	定义双线，双线的宽度等于border-width的值
groove	定义3D凹槽边框。其效果取决于border-color的值
ridge	定义3D菱形边框。其效果取决于border-color的值
inset	定义3D凹边，其效果取决于border-color的值
outset	定义3D凸边，其效果取决于border-color的值

例子：

```
p{
  border-style:outset; /* 3D突边 */
  border-color:gray;
  border-width:8px;
}
span{
  border-style:inset; /* 3D凹边 */
  border-color:gray;
  border-width:8px;
  display:block;
}
```

样式是边框最重要的一个方面，这不是因为样式控制着边框的显示（当然，样式确实控制着边框的显示），而是因为如果没有样式，将根本没有边框。

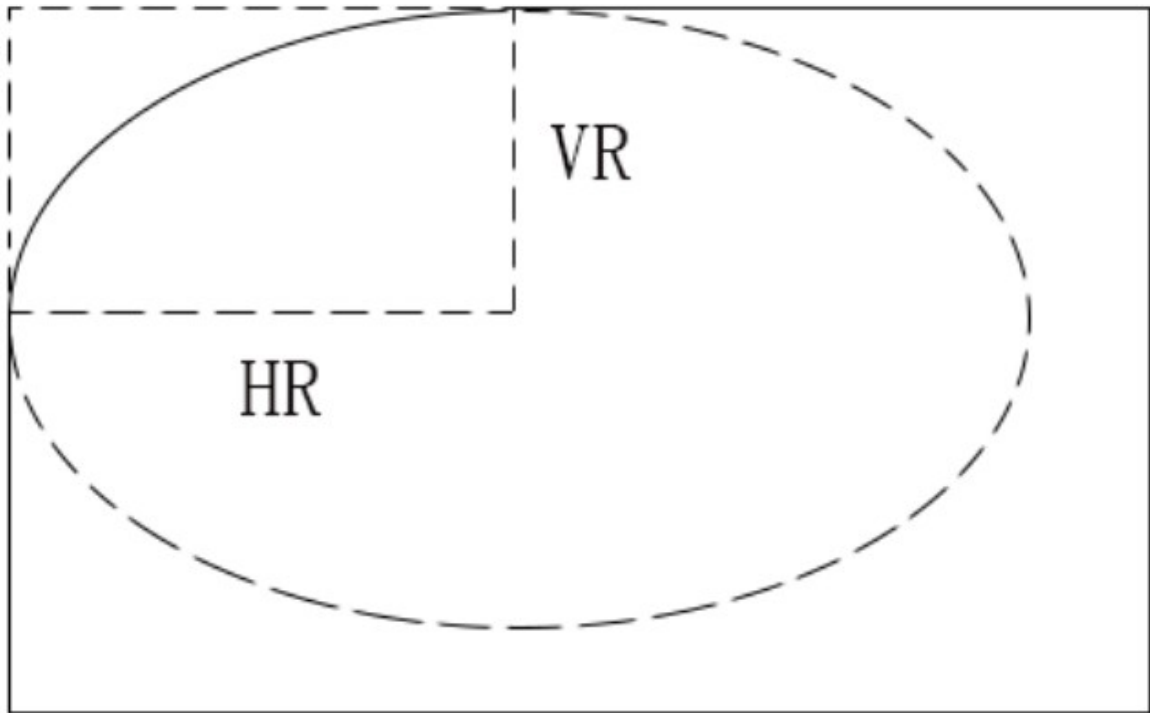
CSS 没有定义 3 个关键字的具体宽度，所以一个用户代理可能把 thin、medium 和 thick 分别设置为等于 5px、3px 和 2px，而另一个用户代理则分别设置为 3px、2px 和 1px。

3.圆角边框

属性：border-radius

圆角边框的圆角是椭圆的一条弧线，如下图所示，HR代表椭圆的水平半径，VR代表椭圆的垂直半径；HR与VR相同时，圆角就会变成四分之一圆弧。

border-radius有四个角进行设置HR和VR



水平半径与垂直半径相等

border-radius有一个参数

```
border-radius: 10px;
```

则元素的top-left、top-right、bottom-right和bottom-left取值均相同。

border-radius有两个参数

```
border-radius: 10px 20px;
```

- **top-left**和**bottom-right**相同(左上和右下)，取值均为属性值1
- **top-right**和**bottom-left**相同(右上和左下)，取值均为属性值2

border-radius有三个参数 (记住右上到左下是一样的)

```
border-radius: 10px 20px 30px;
```

- top-left取值为属性1
- top-right和bottom-left取值属性2
- bottom-right取值属性3

border-radius有四个参数

```
border-radius: 10px 20px 30px 40px;
```

top-left、top-right、bottom-right和bottom-left分别对应4个不同的属性值
即从左上角开始按照顺时针方向依次赋值

水平半径与垂直半径不相等

当水平半径与垂直半径不同时，需要用斜线 (/) 隔开。水平半径（或垂直半径）的参数都也可以是1~4个。

```
border-radius: 10px / 20px; /* 水平 / 垂直 */
```

不同个数参数的赋值顺序与上面相同。

4.边框阴影

属性：box-shadow

```
box-shadow: h-shadow v-shadow [blur] [spread] [color] [inset];
```

值	描述
h-shadow	必须的，用于指定水平阴影的位置，可取负
v-shadow	必须的，用于指定垂直阴影的位置，可取负
blur	可选的，用于指定模糊距离
spread	可选的，用于指定阴影尺寸
color	可选的，用于指定阴影颜色
inset	可选的，将外部阴影(outset)改为内部阴影

h-shadow不同数值对比 10px 与 100px



由此可知阴影是相对于图片进行定位的 正数向右。

模糊距离应该是相对于边框的距离，阴影尺寸是对于图片大小的整体放大(?)

5.图像边框

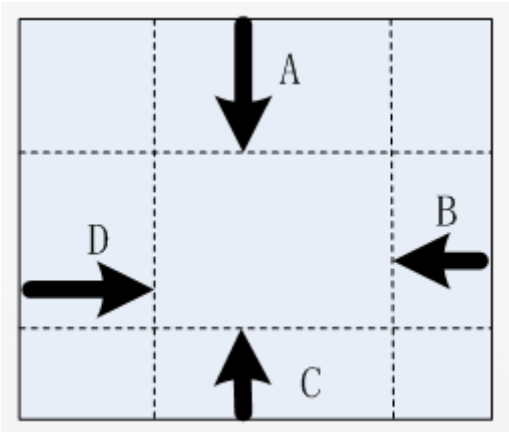
属性: **border-image**

属性	描述
border-image-source	边框的图像的路径
border-image-slice	图像边框向内偏移，从四个方向对边框图片进行切割
border-image-width	设置图像边框的宽度
border-image-repeat	设置图像边框是否应平铺覆盖(repeat)、取整平铺(round)或拉伸覆盖(stretch)
border-image	简写形式，用于把图像边框的所有属性设置到一个声明中

边框背景的分割

属性: **border-image-slice**

```
border-image-slice: 10 20 30 40;  
border-image-slice: 10 20 30 40 fill;
```



- 参数可以有1~4个，允许是数值或百分比
- 参数遵循TRBL原则，按照顺时针方向使用指定宽度对图像进行分割，将图像分割成9部分
- 默认情况下，元素中心区域不填充边框的图像；当提供参数fill时，元素的中心区域将被填充

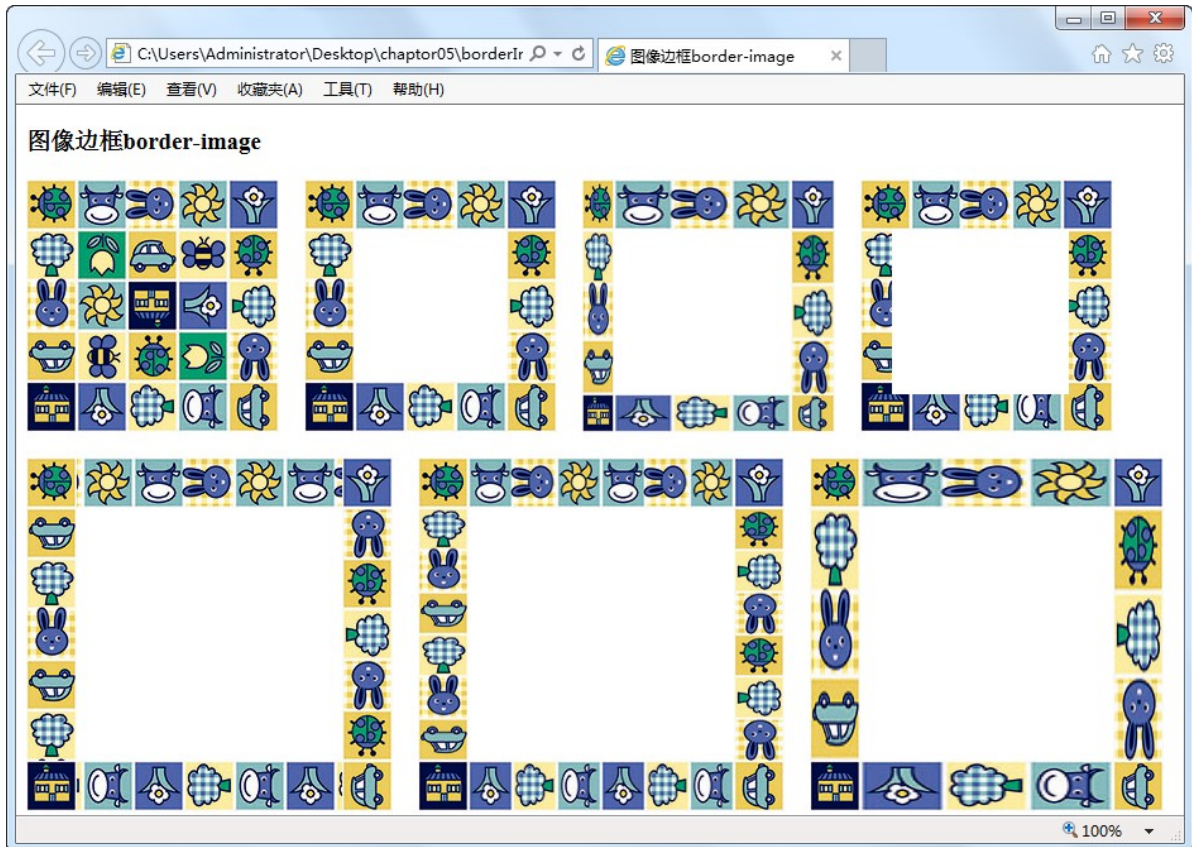
边框背景的使用

属性: **border-image**

```
border-image : url(图像的路径) 图像分割方式(slice)/图像边框宽度(width) 图像平铺方式(repeat)
```

案例:

```
border-image:url("images/borderImage2.jpg") 40;
border-image:url("images/borderImage2.jpg") 40/40px 35px 30px 25px; /*进行了压缩*/
border-image:url("images/borderImage2.jpg") 40 35 30 25/40px 35px 30px 25px; /*等比例的*/
border-image:url("images/borderImage2.jpg") 40/40px repeat;
border-image:url("images/borderImage2.jpg") 40/40px round;
border-image:url("images/borderImage2.jpg") 40/40px stretch;
```



repeat (平铺覆盖)	round (取整平铺)	stretch (拉伸覆盖)
		
以中心点向左右两侧等比例平铺	当最后一个框不能完全显示且显示的区域不到一半，则不显示。反之显示。	水平和垂直方向以拉伸的方式进行显示

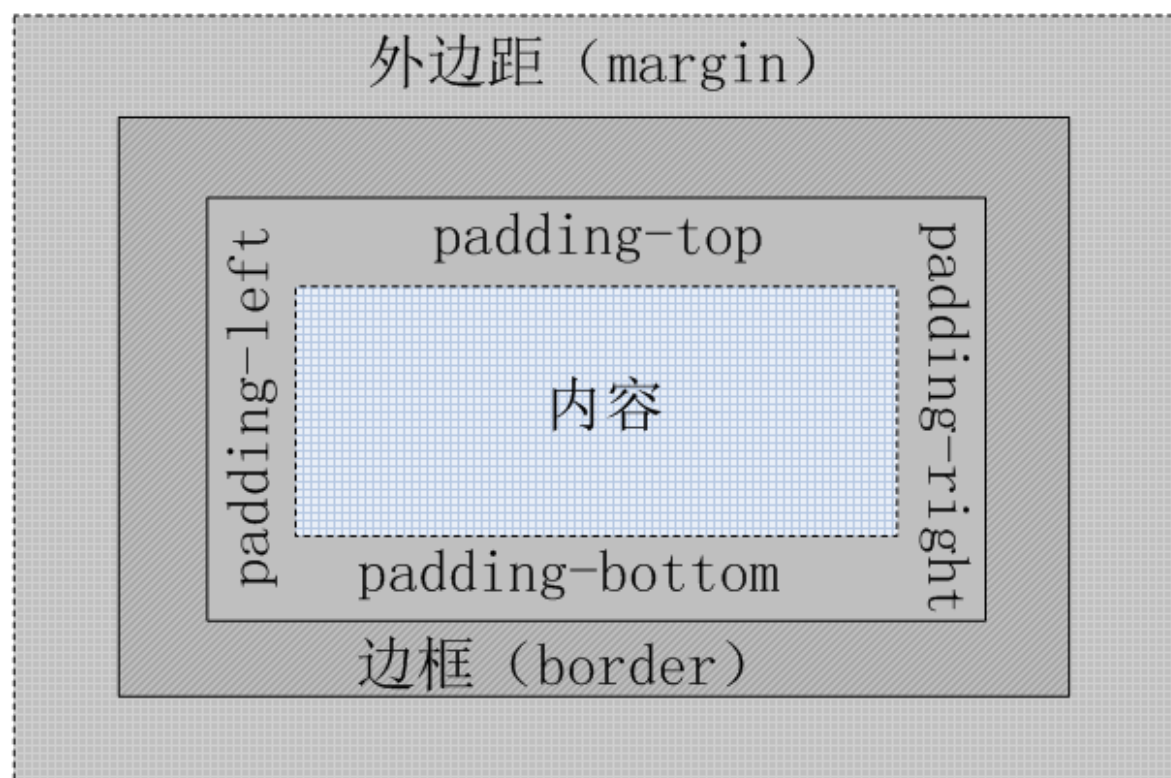
对border-image属性添加fill参数时，该div的中心区域将会被边框图像填充

```
border-image:url("images/borderImage2.jpg") 40 fill/40px stretch;
```




内边距

内边距（padding）是指内容区与边框之间的距离

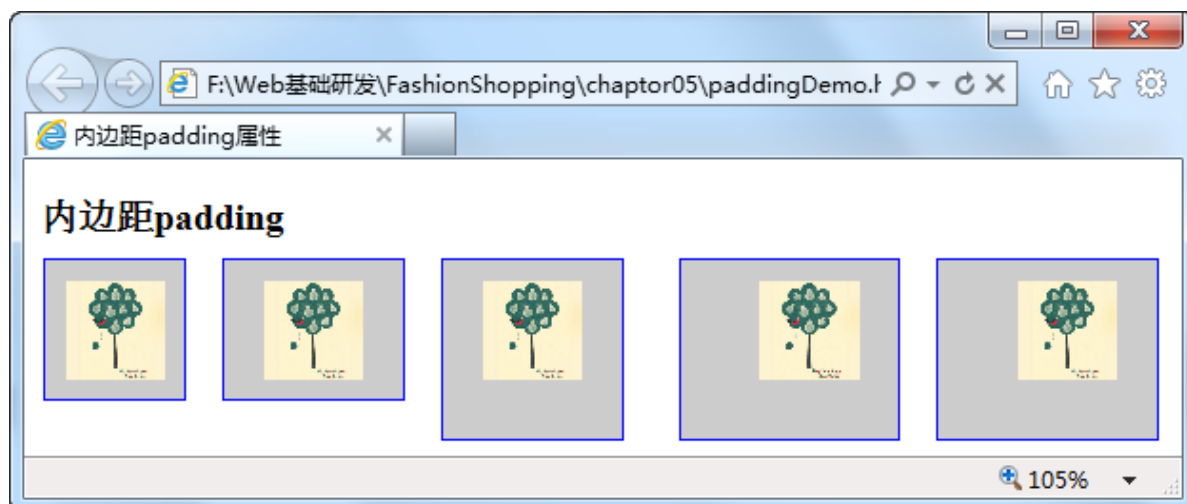


属性：

值	描述
padding-top	设置元素上面的内边距
padding-right	设置元素右侧的内边距
padding-bottom	设置元素下面的内边距
padding-left	设置元素左侧的内边距
padding	简写属性

案例：

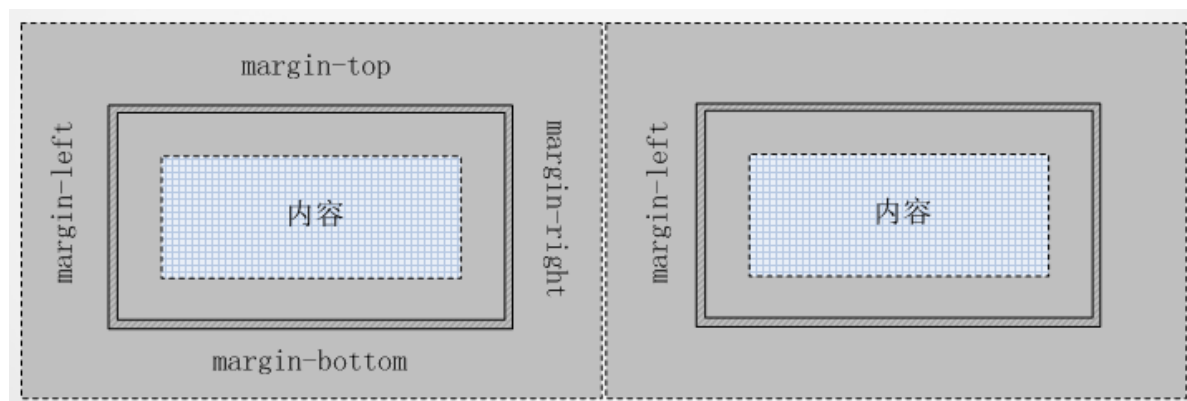
```
padding:10px;
padding:10px 20px;
padding:10px 20px 30px;
padding:10px 20px 30px 40px;
```



可以看到这更像是一种压缩。

外边距

外边距（**margin**）是指元素与元素之间的距离，即围绕在元素边框之外的空白区域，通过外边距可以为元素创建额外的“空间”。



属性：

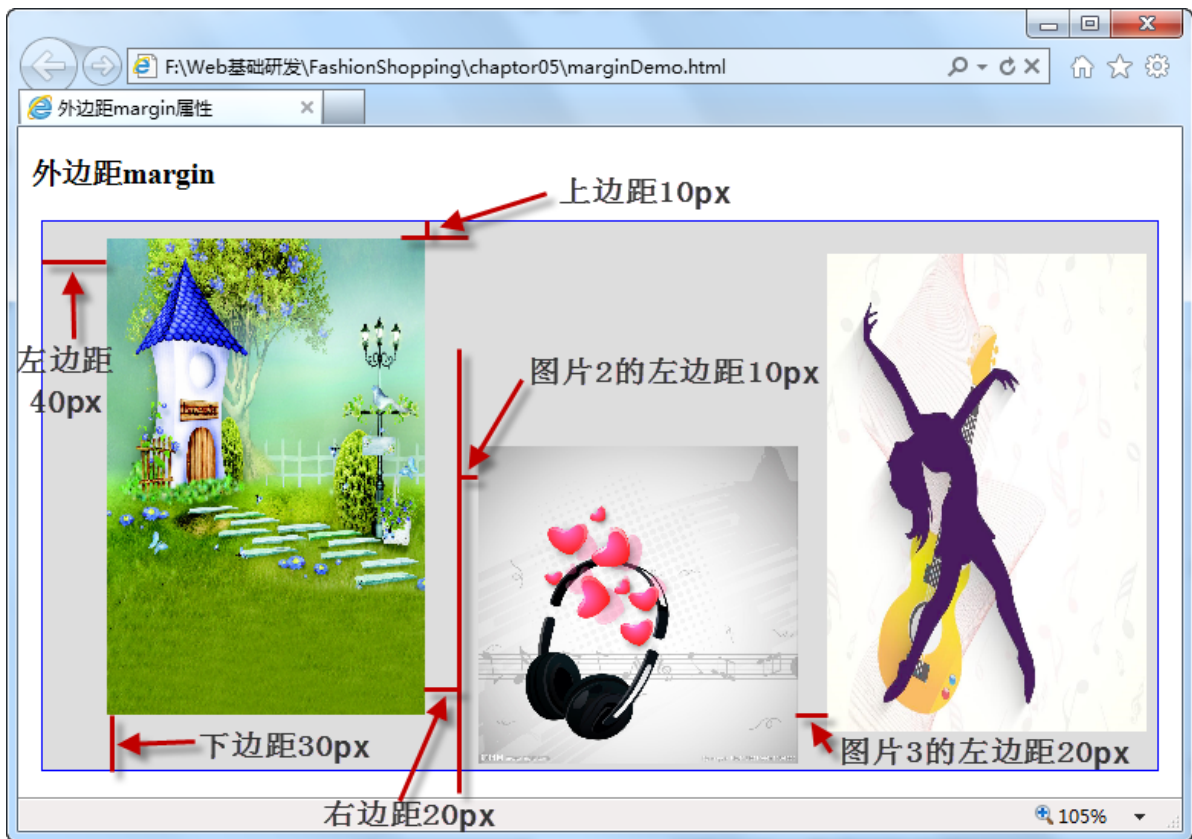
值	描述
margin-top	设置元素上面的外边距
margin-right	设置元素右侧的外边距
margin-bottom	设置元素下面的外边距
margin-left	设置元素左侧的外边距
margin	简写属性

案例：

```

/*第一幅图像*/
margin:10px 20px 30px 40px;
/*第二幅图像*/
margin-left:10px;
/*第三幅图像*/
margin-left:15px;
margin-bottom:20px;

```



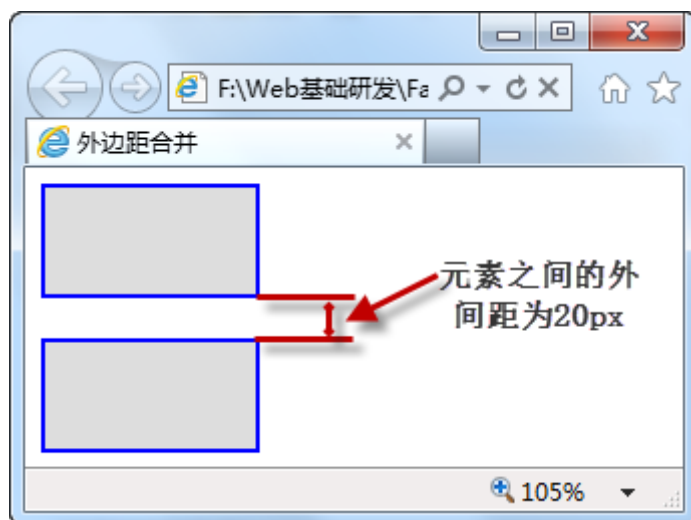
外边距合并

外边距合并（叠加）是指当两个**垂直外边距**相遇时，将形成一个外边距。合并后的外边距的高度，等于合并前的两个外边距中的**较大者**。

1.上下元素间的外边距合并

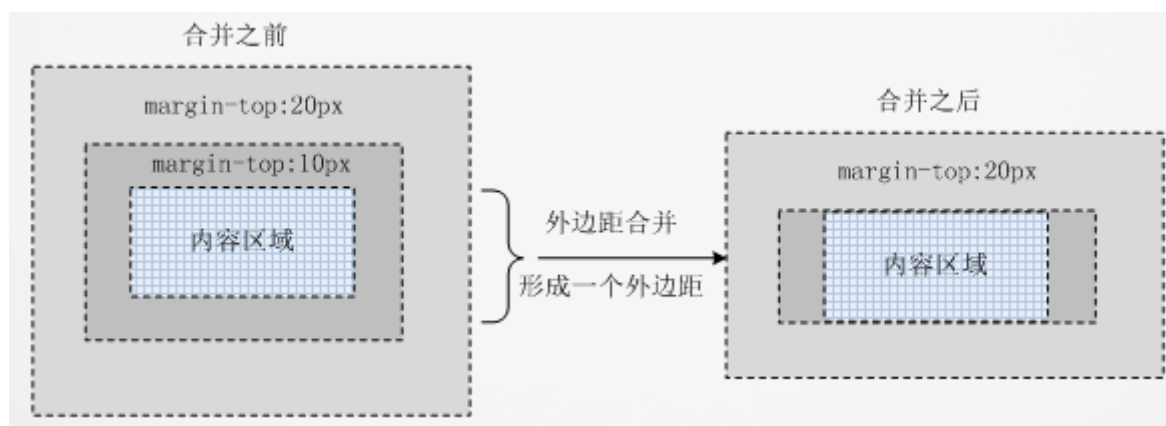
案例：

```
.topDiv{  
    margin-bottom:20px;  
}  
.bottomDiv{  
    margin-top:10px;    /*删去不影响*/  
}
```



2.包含元素间的外边距合并

当一个元素包含在另一个元素中，**父元素没有内边距和边框**，且**子元素没有外边距**时，父元素与子元素的上边距（或下外边距）也会发生合并。



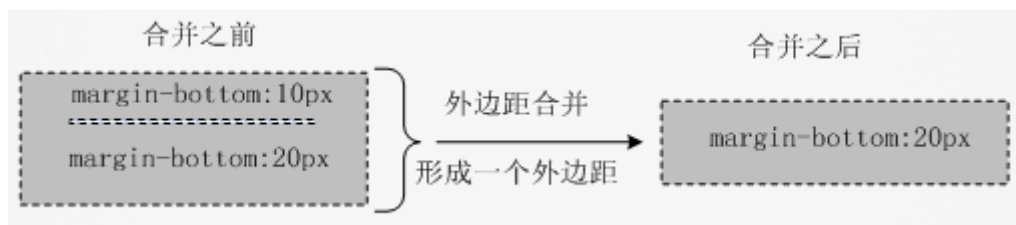
20px是父元素的外边距，10px是子元素的外边距，发生了合并。

解决方案：

- 给父元素定义上边框
- 给父元素定义上内边距
- 添加overflow:hidden（超出的部分隐藏）

3.空元素的外边距合并

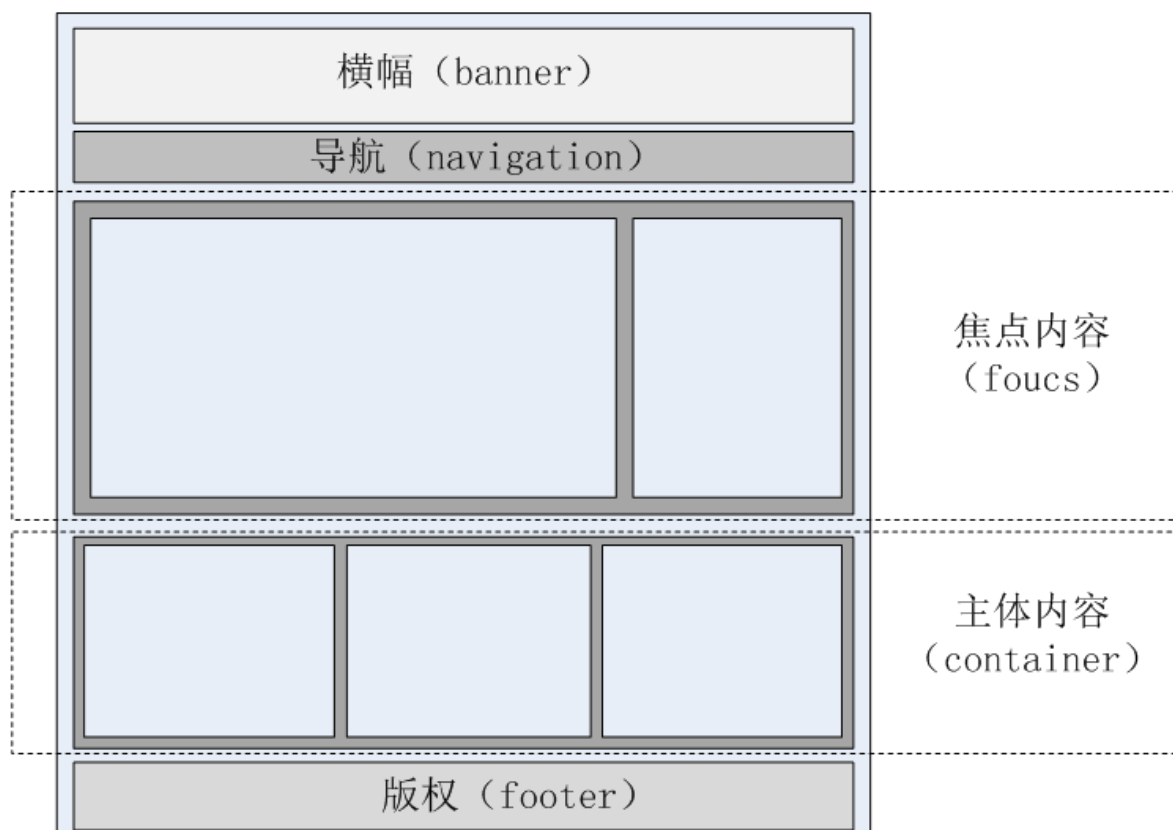
空元素只包含外边距而**无边框和填充**时，上外边距与下外边距就会碰到了一起，元素的上下外边距也会产生合并。



合并后依然可以进行合并操作。

DIV+CSS页面布局

页面布局的核心目标是实现页面的结构与外观相分离，常见的布局方式有三种：**表格布局**、**框架布局**和**DIV+CSS布局**。



———Jerry