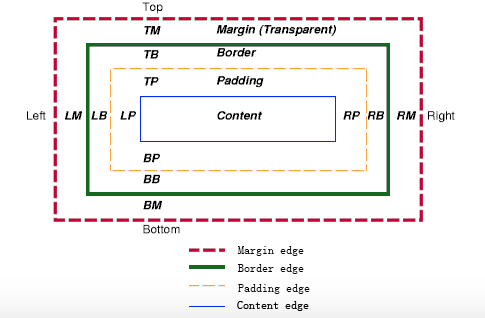
参考：

http://w3help.org/zh-cn/kb/006/

在HTML中，每个元素都会形成一个矩形区域，以便用于定位，这个矩形区域称为框。

元素一般包含4个嵌套的矩形框。



Content边界：由元素的高度和宽度确定的矩形区域。

Padding边界：Padding边界环绕在元素的Padding区域，用以填充背景，即background属性作用在Padding区域和Content区域

Border边界：设置border属性会作用在Border边界，如果border宽度为0，则Border边界与Padding边界重合。

Margin边界：如果margin宽度为0，则margin边界与border边界重合。

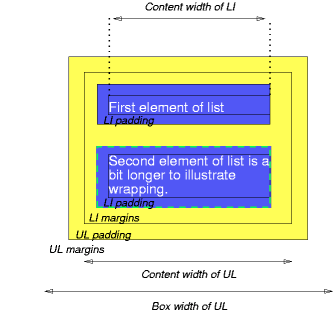
注意：一般情况下，没法看到Content边框和Margin边框。

例子：

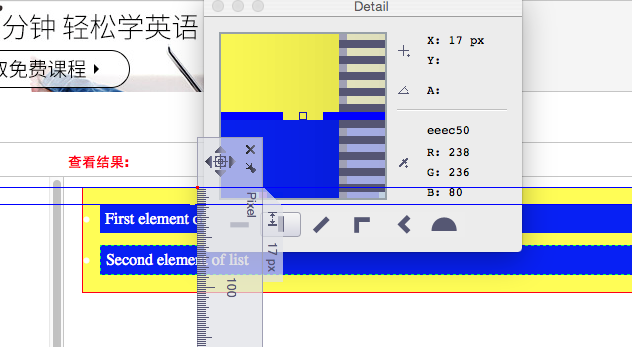
<!DOCTYPEhtml>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<title>第46个CSS页面</title>  
<style>  
  
ul{  
background-color: yellow;  
margin:10px 10px 10px10px;  
padding: 5px 5px5px5px;  
border: 1px solid red;  
}  
  
li {  
color: white;  
background-color: blue;  
margin:12px 12px 12px12px;  
padding:5px 5px 5px5px;  
}  
  
li.l1 {  
border: dashed 1px lime;  
}  
  
</style>  
  
  
</head>  
<body>  
  
<ul>  
<li>  
First element of list  
</li>  
  
<li class="l1">  
Second element of list  
</li>  
</ul>  
  
</body>  
</html>

结果为：

示意图：

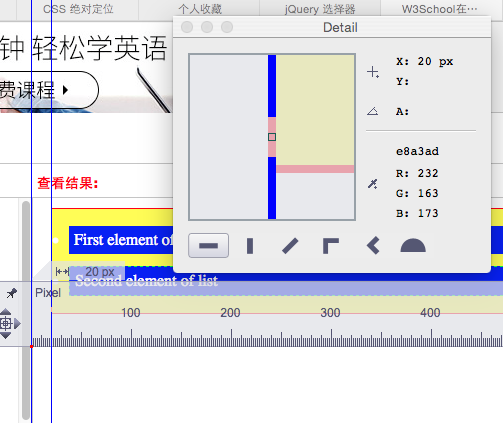


实际运行结果：



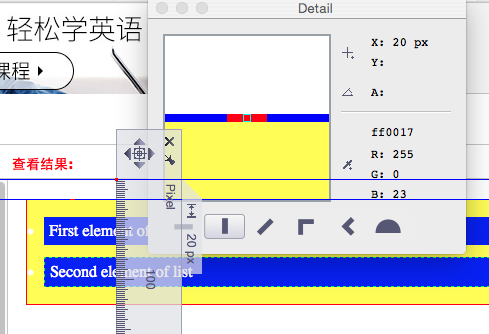
<li>的padding区域到<ul>的padding区域包括li.margin和ul.padding，从图中的测量工具可以看出，<li>蓝色的padding区域到<ul>黄色的padding区域的距离为li.margin+ul.padding正好为17px。

在测试过程中还发现：<ul>的黄色padding区域左边相对浏览器的左边的距离为20px，而设置的margin-left值为10px。将margin-left值设置为0px，测量的距离为10px。

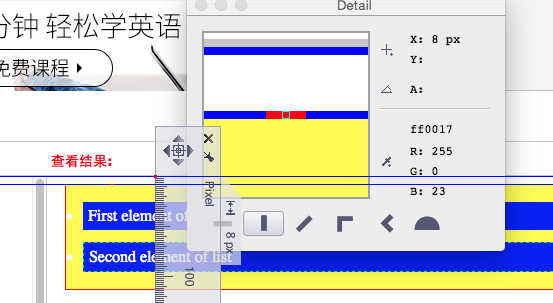


设置<ul>的margin-top的值小于8px时(Netscape 和 IE 对 body 标签定义的默认边距（margin）值是 8px)，测量值为8px，而大于10px时，测量值就为设置的值。

margin-top设置为20px



margin-top设置为5px：



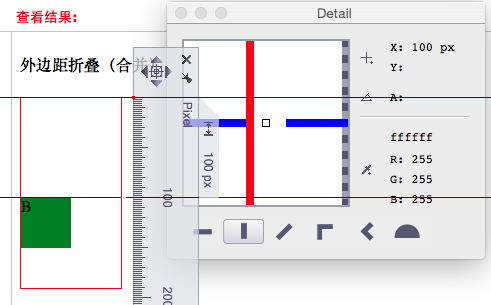
外边距折叠（合并）：

1. 只有垂直外边距才能折叠。
2. 这些外边距没有被非空内容，padding，border或clear分隔开。意思是这些margin之间不能有上述的对象分隔。（有时间元素可能会包含上述对象，但如果这些对象不分隔margin，仍然可以发生折叠）
3. 这些margin必须处于普通文档流中，即非行内框（display:inline-block），非浮动（float属性），非绝对定位（position:absolute）。

注意：margin-top和margin-bottom的区别：

代码如下：

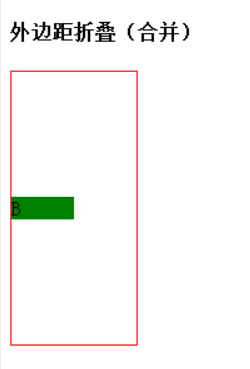
<divstyle="border: 1px solid red; width: 100px;">  
<div style="margin: 40px 0; background-color: green; height: 50px; width: 50px;">  
<div style="margin: 100px 0;">B</div>  
</div>  
</div>



第3个<div>是第2个<div>的子元素，从图中可以看出第2个<div>(背景色为green)的margin-top为40px，第3个<div>（内容为B）的margin-top为100px，发生了外边距折叠，折叠后的margin-top为100px，这里的margin-top是相对于第1个<div>而言。

第2个<div>的margin-bottom并没有和第3个<div>的margin-bottom发生折叠，原因是第2个<div>的height属性不为auto。

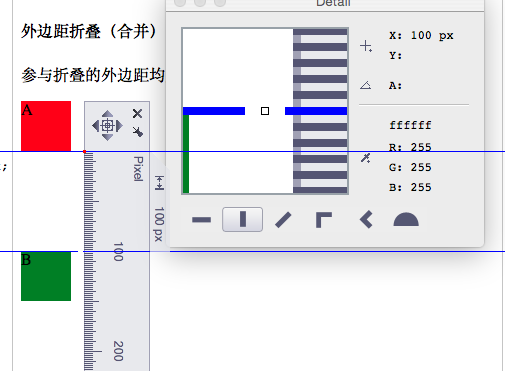
如果去掉第2个<div>的height属性，则默认值为auto，第2个<div>的margin-bottom和第3个<div>的margin-bottom发生折叠。



1. 参与折叠的外边距均为正值，取其中较大的值作为折叠后的外边距

<divstyle="height: 50px; margin-bottom: 50px; width: 50px; background-color: red;">A</div>  
<div style="height: 50px; margin-top: 100px; width: 50px; background-color: green;">B</div>

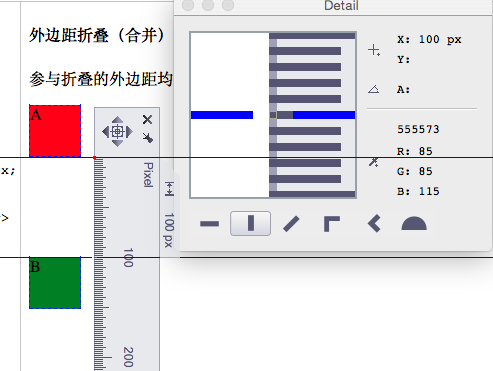
补上加了屏幕测量工具的图。



在这个例子中，如果给两个<div>都加上border，发现外边距仍然发生了折叠。

<divstyle="height: 50px; margin-bottom: 50px; width: 50px; background-color: red; border: 1px dashed blue;">A</div>  
<div style="height: 50px; margin-top: 100px; width: 50px; background-color: green; border: 1px dashed blue;">B</div>

原因是两个<div>虽然包含了border，但是其border没有阻隔第1个<div>的margin-bottom和第2个<div>的margin-top，它们之间仍然是相邻的。



看下面的例子：

未定义border：

<html>

<head>

<style type="text/css">

\* {

margin:0;

padding:0;

border:0;

}

#outer {

width:300px;

height:300px;

background-color:red;

margin-top:20px;

}

#inner {

width:50px;

height:50px;

background-color:blue;

margin-top:10px;

}

</style>

</head>

<body>

<div id="outer">

<div id="inner">

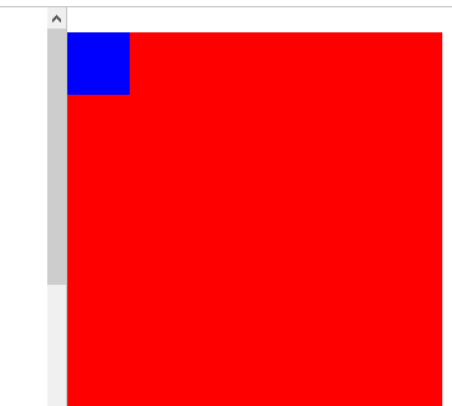
</div>

</div>

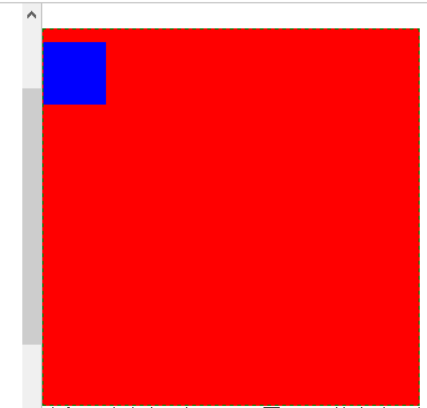
</body>

</html>

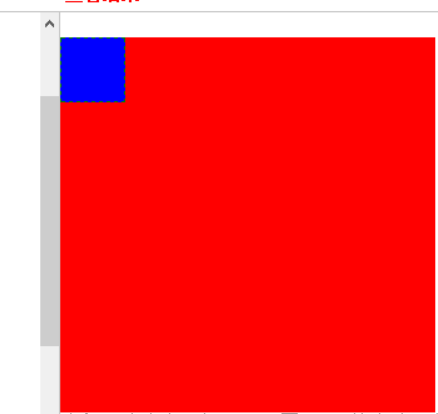
发生外边距折叠：<div>#inner的margin-top与<div>#outer的margin-top发生折叠。



在<div>#outer中添加border：<div>#inner的margin-top不与<div>#outer的margin-top发生折叠，原因是两者之间有border阻隔。



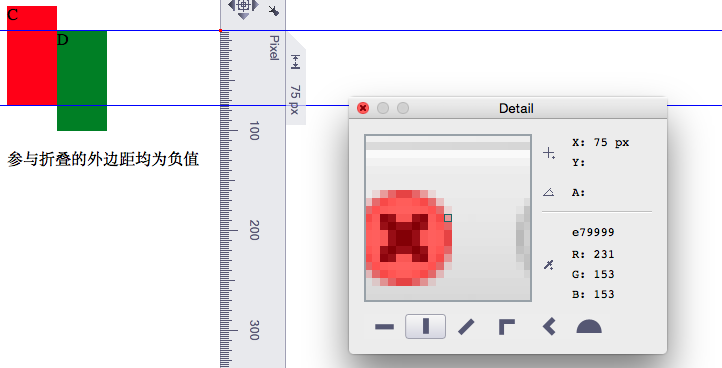
在<div>#inner中添加border：<div>#inner的margin-top与<div>#outer的margin-top仍然发生折叠。原因是border没有阻隔两者，它们仍然是相邻的。



参与折叠的外边距均为负值：

<divstyle="height: 100px; margin-bottom: -75px; width: 50px; background-color: red;">C</div>  
<div style="height: 100px; margin-top: -50px; margin-left: 50px; width: 50px; background-color: green;">D</div>

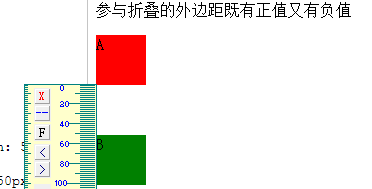
由图中可以看出，参与折叠的外边距为负值时，取绝对值较大的margin值，作为折叠后的margin值，然后从margin=0的位置负向移动。注意：对于第1个<div>(C,red)而言，其margin-bottom=0的位置为其背景色的底边（padding=0），其向下为正，向上为负。对于第2个<div>(D,green)而言，其margin-top=0的位置为其背景色的顶边，其向下为正（原因是有第1个<div>存在），向上为负。



参与折叠的外边距有正有负：先取出负margin中绝对值最大的，然后用其真实值与正margin中最大的相加。

<div style="height: 50px; margin-bottom: -50px; width: 50px; background-color: red;">A</div>  
<div style="height: 50px; margin-top: 100px; width: 50px; background-color: green;">B</div>

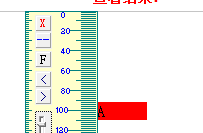
<div>(A,red)的margin-bottom与<div>(B,green)的折叠后的外边距为100+(-50)=50.



相邻的多个margin在计算折叠时，必须放在一起计算而不是分开计算。

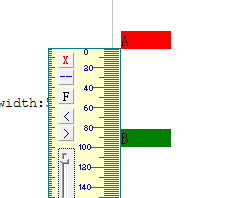
<div style="margin:50px 0; background-color:red; width:50px;">  
 <div style="margin:-60px 0;">  
 <div style="margin:150px 0;">A</div>  
 </div>  
</div>  
  
<div style="margin:-100px 0; background-color:green; width:50px;">  
 <div style="margin:-120px 0;">  
 <div style="margin:200px 0;">B</div>  
 </div>  
</div>

包含<div>(A,red)的3个<div>会发生折叠：折叠后的margin-top为90



而包含<div>(B,green)的3个<div>也会发生折叠，它们之间还会发生折叠，折叠后的为margin-bottom(无法测量出来)。

其中<div>(A,red)和<div>(B,green)之间的外边距折叠是将相邻的6个margin放在一起计算。正margin最大为200，绝对值最大的负margin为-120，所以折叠后的margin为80



创建了块级格式化上下文的元素不会和其子元素发生外边距折叠。

参考：<http://www.cnblogs.com/leejersey/p/3991400.html>

满足下面任一条件，会触发Block Formattin Contexts

1. float除none以外的值。
2. position:absolute,fixed。
3. display:inline-blocks,table-cells,table-captions。
4. overflow:hidden,auto,scroll。

<h4>定义块级格式化上下文的元素</h4>  
<div style="margin-text-outline:50px; width: 100px; height: 100px; background-color: green; overflow:hidden;">  
 <div style="margin-top: 50px; background-color:gold">  
 B  
 </div>  
</div>

第1个<div>没有与第2个<div>发生外边距折叠（如果发生折叠则B应该在green块的最上方）

