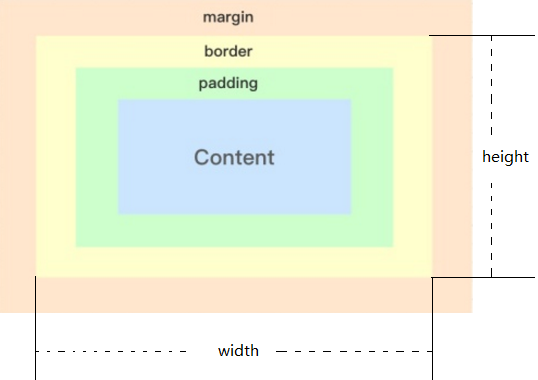
**CSS盒子模型：**

* 内容（CONTENT）就是盒子里装的东西;
* 填充(PADDING)就是怕盒子里装的东西（贵重的）损坏而添加的泡沫或者其它抗震的辅料;
* 边框(BORDER)就是盒子本身了;
* 边界(MARGIN)则说明盒子摆放的时候的不能全部堆在一起，要留一定空隙保持通风，同时也为了方便取出。

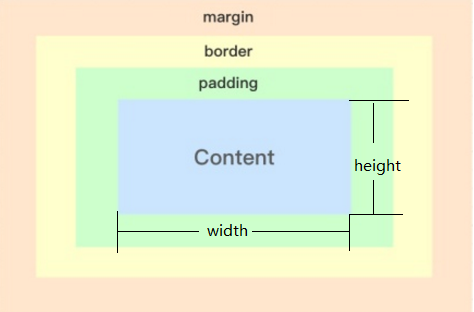


**在怪异模式和标准模式下，浏览器的排版不一样：**

怪异模式：height、width = content + padding + border;



标准模式：height、width = content；



CSS3的新属性（box-sizing）可以改变盒模型的范围：

* border-box：width = border + padding + content；
* content-box：width = content；

**CSS外边距叠加**：在CSS中，两个或多个毗邻的普通流中的盒子（可能是父子元素，也可能是兄弟元素），在垂直方向上的外边距发生叠加。

* 兄弟外边距叠加：上下两个兄弟块级元素，margin是会叠加的，并且以最大的间距为准；

<div class="wrapper"></div>

<div class="wrapper"></div>

.wrapper{

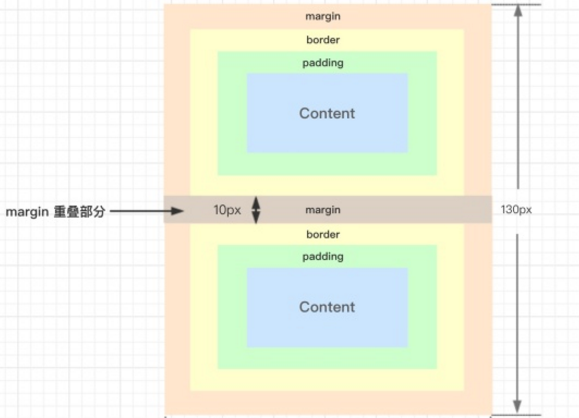
width: 100px;

height: 50px;

margin: 10px;

background-color: #dedede;

}



解决方案：

* + 直接只写一边的margin例如都是margin-top或者都是margin-bottom，我觉得这是我用过最简单的方法了，直接就避免了塌陷。
  + 给两个块状元素形成两个不同的BFC，触发的方法我一般用的是 overflow: hidden;

父子外边距叠加：要求父组件不能设置border或padding值，不能有空余的内容，同时又margin值，块级父元素与其第一个或最后一个子元素（父元素的上外边距和第一个子元素的上外边距；父元素的下外边距和最后一个子元素的下外边距）；

<div class="outer">

<div class="inner"></div>

</div>

.outer{

width: 200px;

height: 100px;

margin: 10px;

background-color: #dedede;

}

.inner{

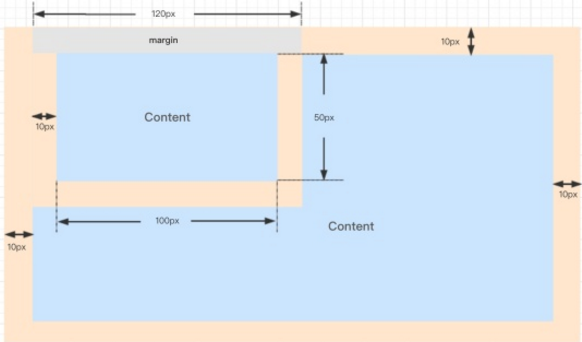
width: 100px;

height: 50px;

margin: 10px;

background-color: #bcbcbc;

}



解决方案：

* + 给父级块级元素设置border；
  + 给父级块级元素设置一个padding；
  + 给父级块级元素触发一个BFC；
* 空块元素：如果存在一个空的块级元素，其 border、padding、inline content、height、min-height都不存在，那么此时上下边距就会发生折叠（自己的上外边距会和自己的下外边距合并）；

#div1,#div2 {

margin:0px;

}

#voidDiv {

margin-top:40px;

margin-bottom:40px;

}

<div id="div1" >这是第一个块级盒子</div>

<div id="voidDiv"></div>

<div id="div2" >这是第二个块级盒子</div> //div1与div2之间的边距是40px而不是80px

解决方案：

* + 设置border
  + 设置padding
  + 设置inline-block等等方法都可

**外边距折叠的计算**一般是以下三种方法:

* 如果两个外边距都为正数，那么取其中较大的数；
* 如果一个为正数一个为负数，那么取它们的代数和；
* 如果两个都为负数，那么取它们其中绝对值大的数；