## 一、BFC是什么？

在解释 BFC 是什么之前，需要先介绍 Box、Formatting Context的概念。

　　Box: CSS布局的基本单位

　　Box 是 CSS 布局的对象和基本单位， 直观点来说，就是一个页面是由很多个 Box 组成的。元素的类型和 display 属性，决定了这个 Box 的类型。 不同类型的 Box， 会参与不同的 Formatting Context（一个决定如何渲染文档的容器），因此Box内的元素会以不同的方式渲染。让我们看看有哪些盒子：

block-level box:display 属性为 block, list-item, table 的元素，会生成 block-level box。并且参与 block fomatting context；

inline-level box:display 属性为 inline, inline-block, inline-table 的元素，会生成 inline-level box。并且参与 inline formatting context；

run-in box: css3 中才有， 这儿先不讲了。

**Formatting context**

　　Formatting context 是 W3C CSS2.1 规范中的一个概念。它是页面中的一块渲染区域，并且有一套渲染规则，它决定了其子元素将如何定位，以及和其他元素的关系和相互作用。最常见的 Formatting context 有 Block fomatting context (简称BFC)和 Inline formatting context (简称IFC)。

　　CSS2.1 中只有 BFC 和 IFC, CSS3 中还增加了 GFC 和 FFC。

**BFC 定义**

　　BFC(Block formatting context)直译为"块级格式化上下文"。它是一个独立的渲染区域，只有Block-level box参与， 它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。

**BFC布局规则：**

内部的Box会在垂直方向，一个接一个地放置。

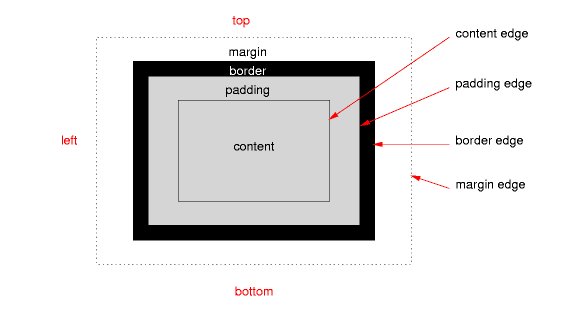
Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠

每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。

BFC的区域不会与float box重叠。

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

计算BFC的高度时，浮动元素也参与计算



**DOM盒模型**

margin、border、padding、content分别定义了元素四种边，然后每种类型的边的四条边定义了一个盒子，分别是content box、padding box、border box、margin box，而决定块盒在包含块中与相邻块盒的垂直间距的便是margin-box，这个margin-box是始终存在的，即使它的margin为0。

**示例：**

.box{

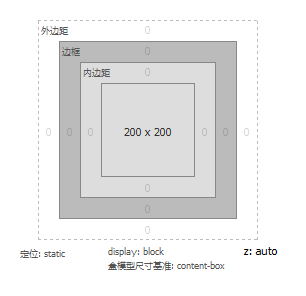
width: 200px;

height: 200px;

background-color: #fff000;

}

<div class="box"></div>



**在BFC中，margin-box会与其相邻的margin-box的对边相折叠。margin-box是参与块级盒在上下文中的布局的，但是参与BFC布局的盒子却是块级盒block-level box，并且还有一点需要明确的是参与布局的是盒子而不是元素本身。**

**整个盒子作为一个 block-level box 参与到页面布局。**

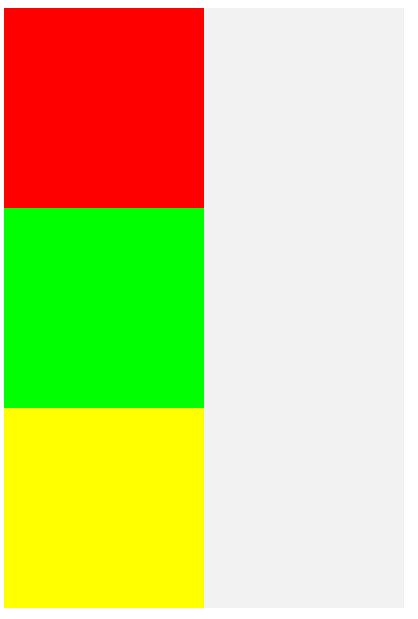
“普通流中的块元素独占一行，然后从上往下一个接一个的排布，相邻元素间会有外边距折叠”

**特殊的例子**

1、浮动元素、绝对定位元素，不是块级盒的块级包含块(比如inline-block、table-cell、table-capation)和overflow值不为visible的块级盒子为它们的内容建立了一个新的块级排版上下文。

2、在一个块级排版上下文中，盒子是从包含块顶部开始，垂直的一个接一个的排列的，相邻两个盒子之间的垂直的间距是被margin属性所决定 的，在一个块级排版上下文中相邻的两个块级盒之间的垂直margin是折叠的。参与BFC的布局的只有普通流normal flow中的块级盒，而float、position值不为relative\static的元素是脱离BFC这一布局环境的，不参与BFC的布局

3、在一个块级排版上下文中，每个盒子的左外边是触碰到包含块的左边的（对于从右向左的排版，则相反），即使在有浮动元素参与的情况下也是如此 (即使一个盒子的行盒是因为浮动而收缩了的)，除非这个盒子新建了一个块级排版上下文(在某些情况下这个盒子自身会因为floats而变窄)。



<div class="stage">

<div class="box box-1 red"></div>

<div class="box box-2 green"></div>

<div class="box box-3 yellow"></div>

</div>

在一个块级排版上下文（**BFC**）中，所有的块级元素按顺序排在盒子容器中，从容器左侧（0，0）位置开始逐个往下堆叠。垂直方向从上往下，水平方向按排版格式确定的方向排列

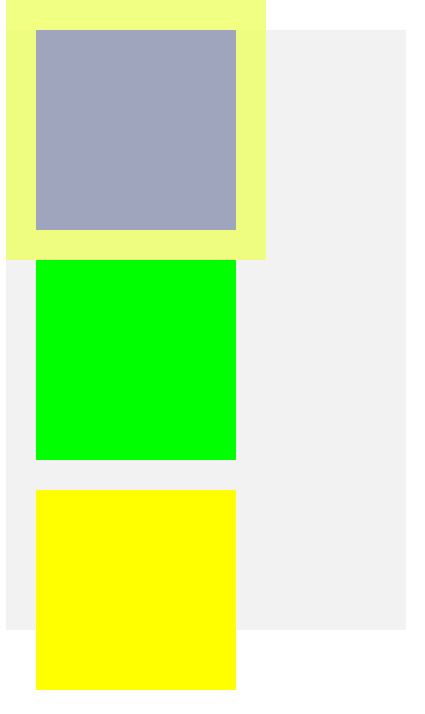
“在一个块级排版上下文中，每个盒子的左外边是触碰到包含块的左边的（对于从右向左的排版，则相反）”

元素盒子的位置和尺寸往往是根据一个矩形计算出来的，我们称这个矩形为元素的包含块。如果这个元素的position值是relative或static，这个元素的包含块是由离其最近的块级的祖先盒子的内容的边content-edge构成的。就是离其最近的块级祖先盒子的content-box。

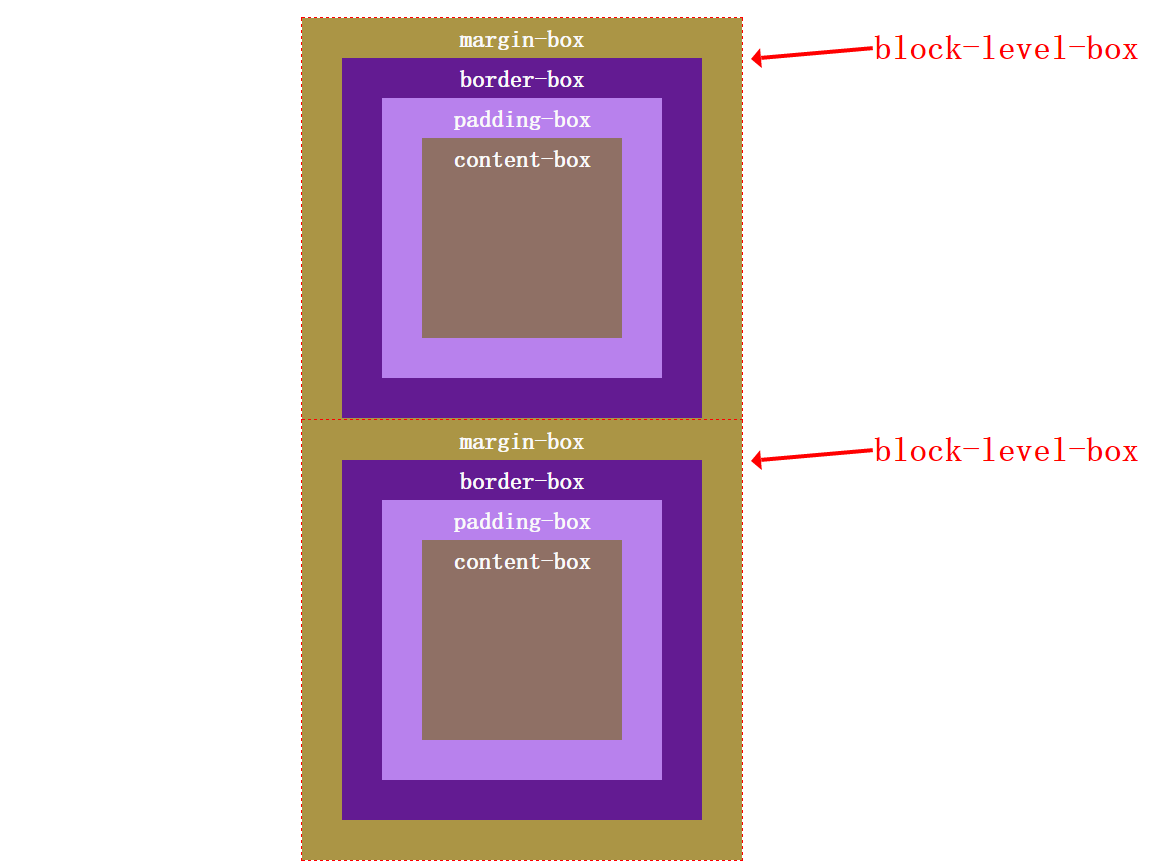
**二、哪些元素会生成BFC?**

* + - 根元素
    - float属性不为none
    - position为absolute或fixed
    - display为inline-block, table-cell, table-caption, flex, inline-flex
    - overflow不为visible

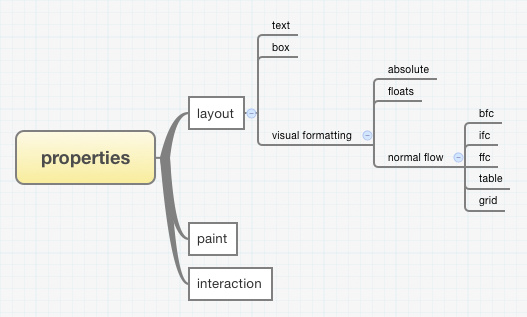
BFC在垂直方向上margin的重合

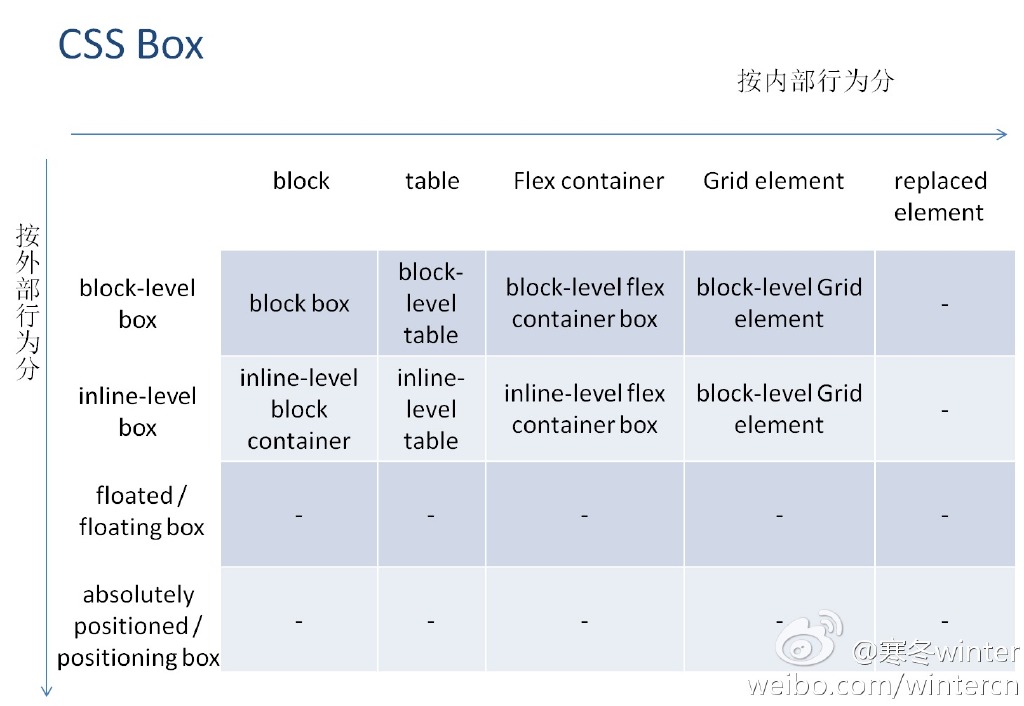


**添加30px的外边距的情况**



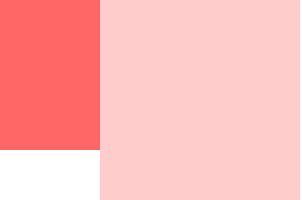
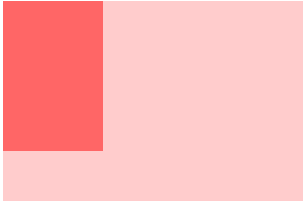
在block-level-box外层构建新的BFC盒子来调整盒子内元素的布局





## 三、BFC的作用及原理

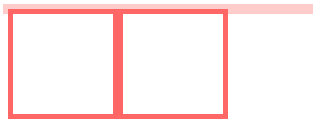
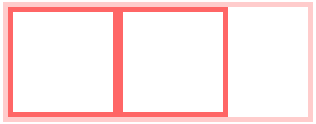
**1. 自适应两栏布局**

****

“每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此“

“BFC的区域不会与float box重叠。”

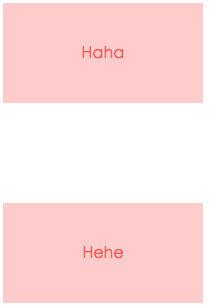
**2. 清除内部浮动**

计算BFC的高度时，浮动元素也参与计算

　　为达到清除内部浮动，我们可以触发par生成BFC，那么par在计算高度时，par内部的浮动元素child也会参与计算。

**3. 防止垂直 margin 重叠**

Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠

**BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。**

**因为BFC内部的元素和外部的元素绝对不会互相影响，因此， 当BFC外部存在浮动时，它不应该影响BFC内部Box的布局，BFC会通过变窄，而不与浮动有重叠。同样的，当BFC内部有浮动时，为了不影响外部元素的布局，BFC计算高度时会包括浮动的高度。避免margin重叠也是这样的一个道理。**