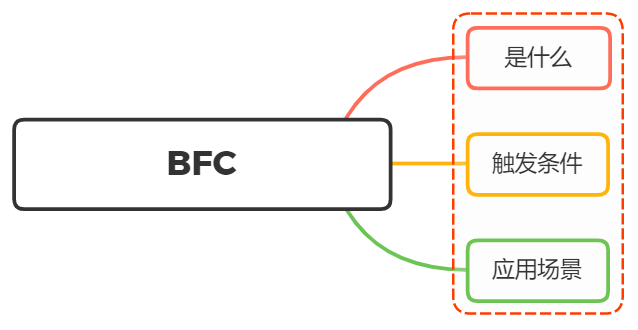
# 面试官：谈谈你对BFC的理解？



## 一、是什么

我们在页面布局的时候，经常出现以下情况：

* 这个元素高度怎么没了？
* 这两栏布局怎么没法自适应？
* 这两个元素的间距怎么有点奇怪的样子？
* ......

原因是元素之间相互的影响，导致了意料之外的情况，这里就涉及到BFC概念

BFC（Block Formatting Context），即块级格式化上下文，它是页面中的一块渲染区域，并且有一套属于自己的渲染规则：

* 内部的盒子会在垂直方向上一个接一个的放置
* 对于同一个BFC的俩个相邻的盒子的margin会发生重叠，与方向无关。
* 每个元素的左外边距与包含块的左边界相接触（从左到右），即使浮动元素也是如此
* BFC的区域不会与float的元素区域重叠
* 计算BFC的高度时，浮动子元素也参与计算
* 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，反之亦然

BFC目的是形成一个相对于外界完全独立的空间，让内部的子元素不会影响到外部的元素

## 二、触发条件

触发BFC的条件包含不限于：

* 根元素，即HTML元素
* 浮动元素：float值为left、right
* overflow值不为 visible，为 auto、scroll、hidden
* display的值为inline-block、inltable-cell、table-caption、table、inline-table、flex、inline-flex、grid、inline-grid
* position的值为absolute或fixed

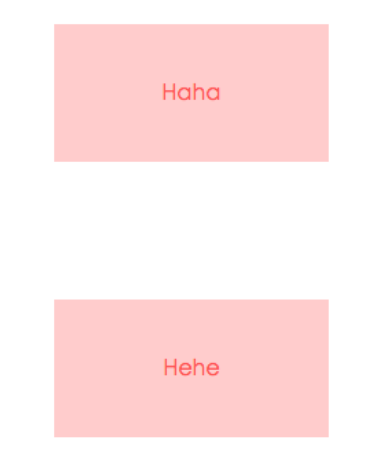
## 三、应用场景

利用BFC的特性，我们将BFC应用在以下场景：

#### 防止margin重叠（塌陷）

<style>  
 p {  
 color: #f55;  
 background: #fcc;  
 width: 200px;  
 line-height: 100px;  
 text-align:center;  
 margin: 100px;  
 }  
</style>  
<body>  
 <p>Haha</p >  
 <p>Hehe</p >  
</body>

页面显示如下：



两个p元素之间的距离为100px，发生了margin重叠（塌陷），以最大的为准，如果第一个P的margin为80的话，两个P之间的距离还是100，以最大的为准。

前面讲到，同一个BFC的俩个相邻的盒子的margin会发生重叠

可以在p外面包裹一层容器，并触发这个容器生成一个BFC，那么两个p就不属于同一个BFC，则不会出现margin重叠

<style>  
 .wrap {  
 overflow: hidden;// 新的BFC  
 }  
 p {  
 color: #f55;  
 background: #fcc;  
 width: 200px;  
 line-height: 100px;  
 text-align:center;  
 margin: 100px;  
 }  
</style>  
<body>  
 <p>Haha</p >  
 <div class="wrap">  
 <p>Hehe</p >  
 </div>  
</body>

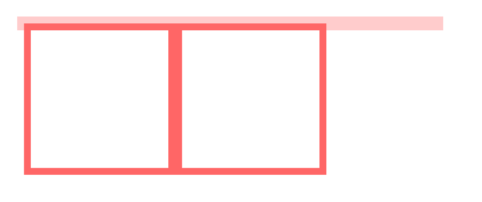
这时候，边距则不会重叠：



#### 清除内部浮动

<style>  
 .par {  
 border: 5px solid #fcc;  
 width: 300px;  
 }  
   
 .child {  
 border: 5px solid #f66;  
 width:100px;  
 height: 100px;  
 float: left;  
 }  
</style>  
<body>  
 <div class="par">  
 <div class="child"></div>  
 <div class="child"></div>  
 </div>  
</body>

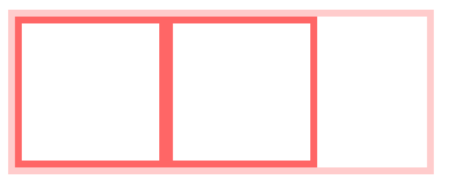
页面显示如下：



而BFC在计算高度时，浮动元素也会参与，所以我们可以触发.par元素生活才能BFC，则内部浮动元素计算高度时候也会计算

.par {  
 overflow: hidden;  
}

实现效果如下：



#### 自适应多栏布局

这里举个两栏的布局

<style>  
 body {  
 width: 300px;  
 position: relative;  
 }  
   
 .aside {  
 width: 100px;  
 height: 150px;  
 float: left;  
 background: #f66;  
 }  
   
 .main {  
 height: 200px;  
 background: #fcc;  
 }  
</style>  
<body>  
 <div class="aside"></div>  
 <div class="main"></div>  
</body>

效果图如下：



前面讲到，每个元素的左外边距与包含块的左边界相接触

因此，虽然.aslide为浮动元素，但是main的左边依然会与包含块的左边相接触

而BFC的区域不会与浮动盒子重叠

所以我们可以通过触发main生成BFC，以此适应两栏布局

.main {  
 overflow: hidden;  
}

这时候，新的BFC不会与浮动的.aside元素重叠。因此会根据包含块的宽度，和.aside的宽度，自动变窄

效果如下：



### 小结

可以看到上面几个案例，都体现了BFC实际就是页面一个独立的容器，里面的子元素不影响外面的元素

## 参考文献

* https://developer.mozilla.org/zh-CN/docs/Web/Guide/CSS/Block\_formatting\_context
* https://github.com/zuopf769/notebook/blob/master/fe/BFC%E5%8E%9F%E7%90%86%E5%89%96%E6%9E%90/README.md