* 一、BFC是什么？
* 二、哪些元素会生成BFC?
* 三、BFC的作用及原理

# BFC是什么？

**盒子模型**

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。

**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)。

**block formatting context布局规则**

* 每个元素会一个紧挨着一个
* 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。（什么情形下是同一个BFC）
* 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。
* 计算BFC的高度时，浮动元素也参与计算
* BFC的区域不会与float box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。

# 二、哪些元素会生成BFC?

* the root element or something that contains it
* floats (elements where [float](https://developer.mozilla.org/en-US/docs/Web/CSS/float) is not none)
* absolutely positioned elements (elements where [position](https://developer.mozilla.org/en-US/docs/Web/CSS/position) is absolute or fixed)
* inline-blocks (elements with [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display): inline-block)
* table cells (elements with [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display): table-cell, which is the default for HTML table cells)
* table captions (elements with [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display): table-caption, which is the default for HTML table captions)
* elements where [overflow](https://developer.mozilla.org/en-US/docs/Web/CSS/overflow) has a value other than visible
* flex boxes (elements with [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display): flex or inline-flex)

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

# 三、BFC的作用及原理

## 1.自适应两栏布局

<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>



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。

因此上面就是那样子的显示。大红色的区域并不会影响粉红色区域的显示。

我们可以通过触发main生成BFC,来实现自适应两栏布局

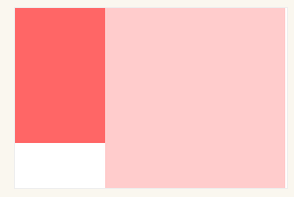
.main {

overflow: hidden;

}

也可以通过float:left这样子的设置来触发。

当触发main生成BFC后，这个新的BFC不会与浮动的aside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：



## 2：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>

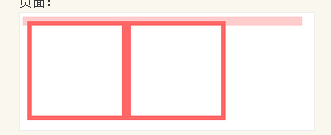
<div class="par">

<div class="child"></div>

<div class="child"></div>

</div>

</body>



使用清除浮动overflow:hidden这种方式，不是很清楚和我们常用的方式有什么区别。

这里是使用的BFC的一条规则

计算BFC的高度时，浮动元素也参与计算

清除浮动有很多种方式吧，但是我并没有常用这种方式。

## 3：防止垂直 margin 重叠

代码：

<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>



p元素默认是display:block.

### marign边界重叠

什么时候会发生margin边界重叠

解决方案

#### 什么时候会发生margin边界重叠

只有普通文档流中块框的垂直边界才会发生边界叠加。行内框、浮动框或绝对定位框之间的边界不会叠加。

备注：

display:inline-block的元素时不会出现margin重叠现象的。

#### 解决方案

外层元素padding代替

内层元素透明边框 border:1px solid transparent;

内层元素绝对定位 postion:absolute:

外层元素 overflow:hidden;

内层元素 加float:left;或display:inline-block;

内层元素padding:1px;

#### 总结

对于margin重叠问题有了进一步的认识，以前以为只要是元素就都会有这个问题（比如display:inline-block 和 float:left）.

这个是个很细节的问题了，自己没有注意到吧。可能

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

我们可以在p外面包裹一层容器，并触发该容器生成一个BFC。那么两个P便不属于同一个BFC，就不会发生margin重叠了。

代码：

<style>

.wrap {

overflow: hidden;

}

p {

color: #f55;

background: #fcc;

width: 200px;

line-height: 100px;

text-align:center;

margin: 100px;

}

</style>

<body>

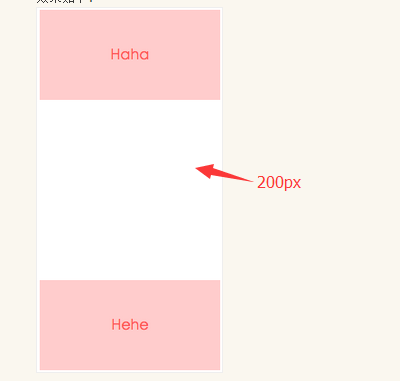
<p>Haha</p>

<div class="wrap">

<p>Hehe</p>

</div>

</body>



参考文献

<http://www.cnblogs.com/lhb25/p/inside-block-formatting-ontext.html>

<http://www.w3.org/TR/CSS21/visuren.html#block-formatting>