

浮动元素与相邻的Block Formatting Context之间的关系

Tags: Block formatting context, float, 浮动

操作系统版本:

Windows 7 Ultimate [ver 6.1.7600.16399] 简体中文版

浏览器版本:

IE6

IE7

IE8

Firefox 3.6.3

Chrome 5.0.366.2 dev

Safari 4.0.5

Opera 10.51

受影响的浏览器:

所有浏览器

1. W3C对Float与Block formatting context之间关系的叙述.....	1
2. 通过overflow特性创建的BFC	2
2.1. 容器的剩余宽度大于BFC的宽度	2
2.2. 容器的剩余宽度小于BFC的宽度，BFC的宽度小于容器的宽度	2
2.2.1. Firefox中的特殊现象	3
2.3. BFC的宽度大于容器的宽度	4
3. 此现象可能导致的问题	5
4. 如何避免此现象的发生	5

1. W3C对Float与Block formatting context之间关系的叙述

[W3C CSS Level 2 Revision 1 第9.5节](#)关于浮动的叙述中对Float与Block formatting context之间的关系做了如下描述:

表格的"border box"、块级的替换元素、或是在普通流中创建了新的"block formatting context" (如元素的"overflow"特性不为"visible"时)的元素不可以与位于相同的"block formatting context"中的浮动元素相重叠。如果有必要,布局实现中应清除上述元素而将其放置在之前的所有浮动元素之下的一行,但也可能在有足够剩余空间的情况下与浮动元素相邻。它们甚至可能会使上述元素的"border box"比根据普通流中的非替换块级元素宽度算法计算而来的更窄。CSS2并没有定义用户端何时将上述元素紧邻浮动元素放置或是变窄。

在[第9.4.1节](#)中,说明了何种情况下会创建"block formatting context":

- 浮动元素 (float:left|right);
- 绝对定位元素 (position:absolute|fixed);
- 行内块元素 (display:inline-block);
- 表格的单元格 (display:table-cells, TD, TH);
- 表格的标题 (display:table-captions, CAPTION);
- "overflow"特性不为"visible"的元素 (除非该值已经传播到视口)。

另外在[第17.4节](#)关于表格可视化模型中还提到:

- 表格元素创建的“匿名框”也会创建新的"block formatting context"。

可见，CSS规范中仅提到了"block formatting context"与浮动元素直接“不可重叠”，并没有详细描述如何处理它们之间的布局关系。

【注】由于IE6/7与IE8(Q)中没有"block formatting context"的概念，取而代之的是hasLayout特性。关于hasLayout与"block formatting context"直接的关系参见：[比较IE的hasLayout特性和CSS规范中的block formatting context](#)。

为方便描述，以下段落将"block formatting context"以"BFC"代替。

2. 通过overflow特性创建的BFC

2.1. 容器的剩余宽度大于BFC的宽度

当容器有足够的剩余空间容纳BFC的宽度时，所有浏览器都会将BFC放置在浮动元素所在行的剩余空间内。

测试样例：

```
<style>
* { font:12px Arial; }
.container { background:silver; width:150px; height:30px; }
.float { background:darkgray; float:left; width:80px; }
.bfc { background:dimgray; overflow:hidden; color:white; }
</style>
<div class="container">
  <div class="float">FLOAT(80)</div>
  <div class="bfc" style="width:60px;">BFC(60)</div>
</div>
```

测试代码中容器【container】的宽度为150px，内部包含一个左浮动宽度为80px的元素【float】，及一个设置了overflow:hidden的宽度60px的元素【bfc】，则【bfc】由于overflow特性不为visible而创建了新的BFC。在各浏览器中效果如下：

IE6/7/8, Firefox, Chrome, Safari, Opera

FLOAT(80) BFC(60)

可见，当容器的剩余宽度足够容纳BFC的宽度时，各浏览器的处理相同，BFC被放置到了浮动元素所在行的剩余空间内。

【注】IE6及IE7/8(Q)出现的“3px Bug”不是本文讨论的重点，由于这个Bug触发的差异在本文中将被忽略。

2.2. 容器的剩余宽度小于BFC的宽度，BFC的宽度小于容器的宽度

在IE6/7/8、Chrome、Opera中，当BFC的宽度介于“容器剩余宽度”与“容器宽度”之间时，BFC会显示在浮动元素的下一行；在Safari中，BFC则仍然保持显示在浮动元素所在行，并且BFC溢出容器；在Firefox中，当容器本身也创建了BFC或者容器的padding-top、border-top-width不都为0时表现与IE8(S)、Chrome类似，否则表现与Safari类似。

测试样例：

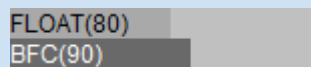
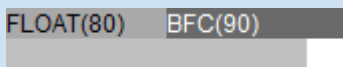
```
<style>
* { font:12px Arial; }
```

```

        .container { background:silver; width:150px; height:30px; }
        .float { background:darkgray; float:left; width:80px; }
        .bfc { background:dimgray; overflow:hidden; color:white; }
    </style>
    <div class="container">
        <div class="float">FLOAT(80)</div>
        <div class="bfc" style="width:90px;">BFC(90)</div>
    </div>

```

测试代码中容器【container】的宽度为150px，内部包含一个左浮动宽度为80px的元素【float】，及一个设置了 overflow:hidden的宽度90px的元素【bfc】，则【bfc】由于overflow特性不为visible而创建了新的BFC。在各浏览器中效果如下：

IE6/7/8, Chrome, Opera	Firefox, Safari
	

可见，当容器的剩余宽度不够容纳BFC的宽度，但BFC的宽度小于容器的宽度时，各浏览器的处理产生差异：

- 在IE6/7/8、Chrome、Opera中，虽然BFC的宽度不足以使其显示在浮动元素所在行，但其宽度小于容器宽度，所以显示到了浮动元素的下一行；
- 在Firefox、Safari中，浏览器仍然将BFC显示在浮动元素所在行，同时BFC溢出容器。

2.2.1. Firefox中的特殊现象

测试样例：

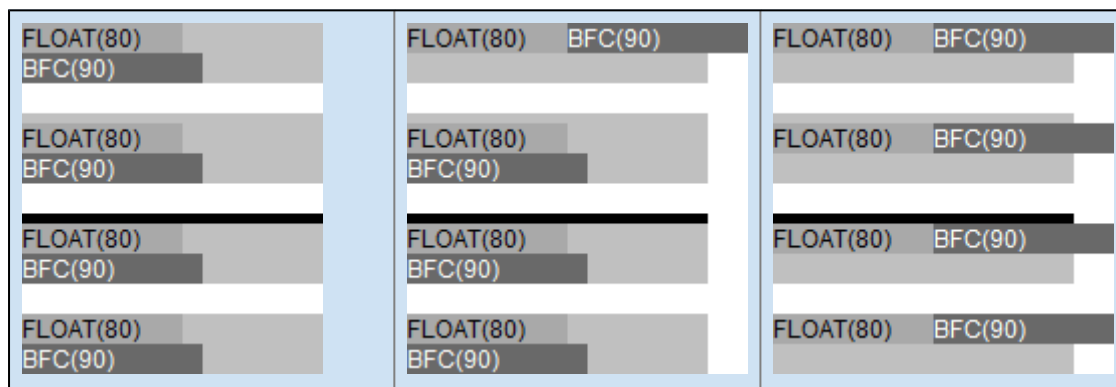
```

<style>
    * { font:12px Arial; }
    .container { background:silver; width:150px; height:30px; }
    .float { background:darkgray; float:left; width:80px; }
    .bfc { background:dimgray; overflow:hidden; color:white; }
</style>
<div class="container">
    <div class="float">FLOAT(80)</div>
    <div class="bfc" style="width:90px;">BFC(90)</div>
</div>
<br />
<div class="container" style="padding-top:5px;">
    <div class="float">FLOAT(80)</div>
    <div class="bfc" style="width:90px;">BFC(90)</div>
</div>
<br />
<div class="container" style="border-top:5px solid black;">
    <div class="float">FLOAT(80)</div>
    <div class="bfc" style="width:90px;">BFC(90)</div>
</div>
<br />
<div class="container" style="display:inline-block;">
    <div class="float">FLOAT(80)</div>
    <div class="bfc" style="width:90px;">BFC(90)</div>
</div>

```

测试代码分为4组，第一组与2.2中的代码相同，第二组为【container】添加了padding-top:5px，第三组为【container】添加了border-top:5px solid black，第四组为【container】添加了display:inline-block。各浏览器中效果如下：

IE6/7/8, Chrome, Opera	Firefox	Safari
------------------------	---------	--------



可见，容器的padding-top不为0、border-top-width不为0、以及容器本身也创建了BFC后，Firefox中对于【BFC】宽度介于“容器剩余宽度”与“容器宽度”之间的情况下，【BFC】会显示到浮动元素的下一行。

2.3. BFC的宽度大于容器的宽度

在IE6/7/8、Opera中，当BFC的宽度大于“容器宽度”时，BFC会显示在浮动元素的下一行；在Chrome、Safari中，BFC则仍然保持显示在浮动元素所在行，并且BFC溢出容器；在Firefox中，当容器本身也创建了BFC或者容器的padding-top、border-top-width不都为0时表现与IE8(S)类似，否则表现与Chrome类似。

测试样例：

```
<style>
  * { font:12px Arial; }
  .container { background:silver; width:150px; height:30px; }
  .float { background:darkgray; float:left; width:80px; }
  .bfc { background:dimgray; overflow:hidden; color:white; }
</style>
<div class="container">
  <div class="float">FLOAT(80)</div>
  <div class="bfc" style="width:151px;">BFC(151)</div>
</div>
<br />
<div class="container" style="padding-top:5px;">
  <div class="float">FLOAT(80)</div>
  <div class="bfc" style="width:151px;">BFC(151)</div>
</div>
<br />
<div class="container" style="border-top:5px solid black;">
  <div class="float">FLOAT(80)</div>
  <div class="bfc" style="width:151px;">BFC(151)</div>
</div>
<br />
<div class="container" style="display:inline-block;">
  <div class="float">FLOAT(80)</div>
  <div class="bfc" style="width:151px;">BFC(151)</div>
</div>
```

测试代码分为4组，与2.2.1中的区别为【BFC】的宽度变为了151px。各浏览器中效果如下：

IE6/7/8, Opera	Firefox	Chrome, Safari
----------------	---------	----------------

FLOAT(80) BFC(151)	FLOAT(80) BFC(151)	FLOAT(80) BFC(151)
FLOAT(80) BFC(151)	FLOAT(80) BFC(151)	FLOAT(80) BFC(151)
FLOAT(80) BFC(151)	FLOAT(80) BFC(151)	FLOAT(80) BFC(151)
FLOAT(80) BFC(151)	FLOAT(80) BFC(151)	FLOAT(80) BFC(151)

可见，

- 在IE8(S)、Opera中，【BFC】的宽度超出了容器剩余宽度，同时也超出了容器的宽度，浏览器将【BFC】显示到了浮动元素的下一行，并且【BFC】溢出容器；
- 在Chrome、Safari中，浏览器发现无论将【BFC】放置在浮动元素所在行还是浮动元素的下一行均会出现【BFC】溢出容器现象，最终将【BFC】显示在浮动元素所在行的剩余空间内，同时溢出容器；
- 在Firefox中，与2.2.1节类似，容器的padding-top不为0、border-top-width不为0、以及容器本身也创建了BFC后，浏览器将【BFC】显示到了浮动元素的下一行，否则显示到浮动元素所在行。

【注】IE6及IE7/8(Q)出现的“[width/height的设定值可能被其内容撑大](#)”不是本文讨论的重点，由于这个Bug触发的差异在本文中将被忽略。

3. 此现象可能导致的问题

若容器的宽度、浮动元素的宽度、BFC的宽度的值设置不当，导致BFC的宽度大于“容器的剩余空间宽度”时，在各浏览器直接可能会触发兼容性问题，最终造成布局上的极大的差异，如内容丢失、错位等。

4. 如何避免此现象的发生

合理地设置容器的宽度、浮动元素的宽度、BFC的宽度的值，尽量保证BFC的宽度小于“容器的剩余空间宽度”。若需要BFC折行显示在新的一行上，可以通过BFC设置“clear”等手段使其换行。