对CSS中的Position、Float属性的一些深入探讨 - CoffeeDeveloper

HTML布局的基本要点：

如果要掌握、运用好Position、Float属性必须要对HTML的两个基本点有清晰的了解。

1. 盒子模型（box model）
2. HTML的普通流（normal flow）

盒子模型

在HTML中元素的盒子模型分为两种：块状元素、行内元素，请注意这里的块状元素（Block）和行内元素（Inline）与Display属性中的inline、block两个属性值并不等同。盒子模型中的Inline、Block类似于是Display属性的父类，例如：Display属性中的list-item属性值是属于块状（Block）类型的。

我们直观的上看两种盒子模型的区别

- 块状（Block）类型的元素可以设置width、height属性，而行内（Inline）类型设置无效。
- 块状（Block）类型的元素会独占一行（直观的说就是会换行显示，无法与其他元素在同一行内显示，除非你主动修改元素的样式），而行内（Inline）类型的元素则都会在一行内显示。
- 块状（Block）类型的元素的width默认为100%，而行内（Inline）类型的元素则是根据自身的内容及子元素来决定宽度。

列举出一些大家常见的元素的分类

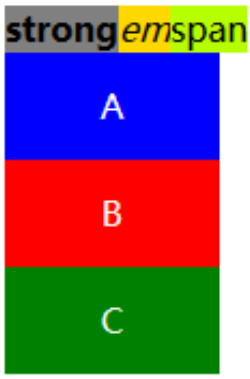
- 块状元素：P、DIV、UL、LI、DD、DT...
- 行内元素：A、IMG、SPAN、STRONG...

HTML的普通流

浏览器在读取HTML源代码的时候是根据元素在代码出现的顺序读取，最终元素的呈现方式是依据元素的盒子模型来决定的。行内元素是从左到右，块状元素是从上到

下。（如下图）

```
<style type="text/css">
  div { width: 100px; height: 50px; line-height: 50px; text-
align: center; color: #fff; }
  strong { background: #808080; }
  em { background: #ffd800; }
  span { background: #b6ff00; }
</style>
<strong>strong</strong><em>em</em><span>span</span>
<div style="background: blue">A</div>
<div style="background: red">B</div>
<div style="background: green">C</div>
```



如果你不改变元素的默认样式前提下，元素在HTML的普通流中会“占用”一个位置，而“占用”位置的大小、位置则是由元素的盒子模型来决定。因此，在后续讲的Position、Float属性**是否会使元素脱离这个普通流是一个关键点**。

Position属性：

我们首先来谈谈Position属性，因为Position属性能够很好的体现HTML普通流这个特征。重点在于应用了不同的position值之后是否有脱离普通流和改变Display属性这两点。

Position属性值

Position的属性值共有四个static、relative、absolute、fixed。

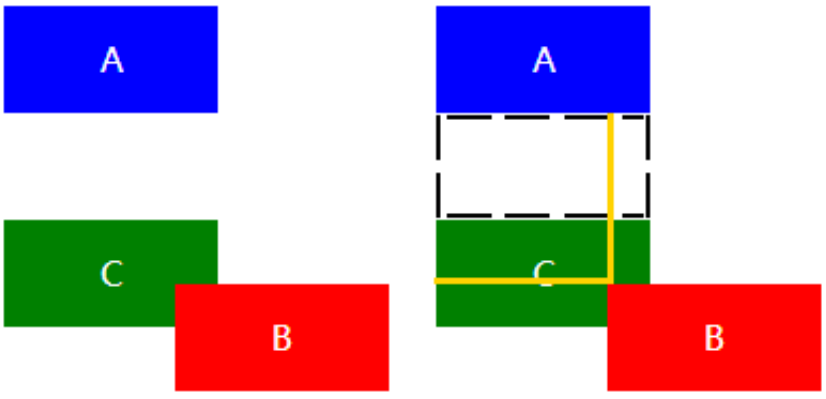
Static

所有元素在默认的情况下position属性均为static，而我们在布局上经常会用到的相对定位和绝对定位常用的属性top、bottom、left、right在position为static的情况下无效。其用法为：在改变了元素的position属性后可以将元素重置为static让其回归到页面默认的普通流中。

Relative

俗称的相对定位，重点在于对**相对**理解。我们此前说过每个元素在页面的普通流中会有“占用”一个位置，这个位置可以理解为默认位置，而相对定位就是将元素偏离元素的默认位置，**但普通流中依然保持着原有的默认位置，并没有脱离普通流，只是视觉上发生的偏移。**
我们先用块状元素来做个示例：

```
<style type="text/css">
div{ width: 100px; height: 50px; line-height: 50px; text-
align: center; color: #fff; }
</style>
<div style="background: blue">A</div>
<div style="background: red; position: relative; top: 20px;
left: 20px;">B</div>
<div style="background: green">C</div>
```

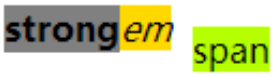


右图中的黑色虚线部分为元素B的默认普通流位置，而黄色线则代表元素B的相对偏移量。我们可以很明显的看出在元素C依然还是保留在原位，并没有因为元素B发生了偏

移而随之变化。

我们再来看看行内元素（在这里用大家最常用的span来做示例）

```
<style type="text/css">
strong { background: #808080; }
em { background: #ffd800; }
span { background: #b6ff00; position: relative; top: 10px;
left: 10px; width: 100px; }
</style>
<strong>strong</strong><em>em</em><span>span</span>
```



请注意看，在这里我是有对span进行width属性的赋值（为100px）。但是我们可以看到span在运用了relative这个position属性值后，依然对width属性无效，换而言之，**position: relative并没有改变行内元素的Display属性，这个概念非常重要（注意与接下来的absolute的区别）**。

Absolute

俗称的绝对定位，绝对定位是相对而言的，怎么理解呢？应用了position: absolute的元素会循着节点树中的父（祖）元素来确定“根”，然后相对这个“根”元素来偏移。如果在其节点树中所有父（祖）元素都没有设置position属性值为relative或者absolute则该元素最终将对body进行位置偏移。应用了position: absolute的元素会**脱离页面中的普通流并改变Display属性（重点）**！

我们先用一个默认嵌套的DIV来做示例

```
<body style="background: yellow;">
    <div style="background: #fff">
        A
        <div style="background: #81b6ff">
```

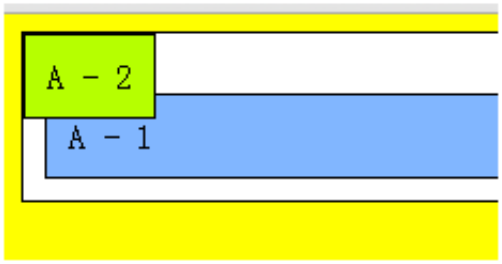
```
            A - 1
            <div style="background: #b6ff00;">
                A - 2
            </div>
        </div>
    </div>
</body>
```

现在我们对A-2这个div设置绝对定位(Top: 0, Left: 0)，而没有对它的父元素（A、A-1）设置任何的position值

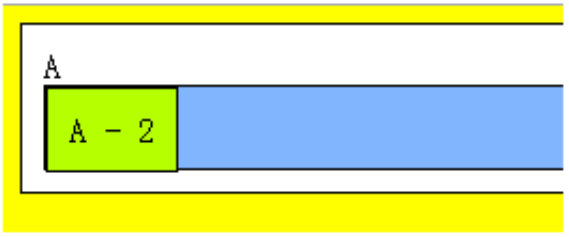
```
<body style="background: yellow;">
    <div style="background: #fff">A
        <div style="background: #81b6ff">A - 1
            <div style="background: #b6ff00; position:
absolute; top: 0; left: 0;">
                A - 2</div></div></div></body>
```

可以看到（A-2）最终是根据body来产生了位移，让我们对比分别设置一下父元素position。

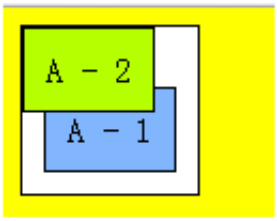
A 为 position: relative 的情况



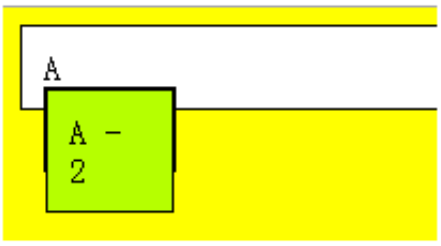
A - 1 为 position: relative 的情况



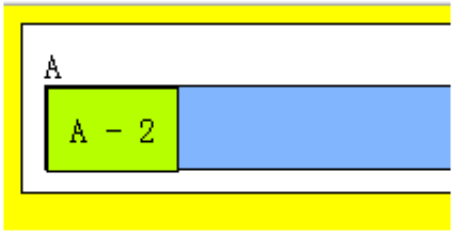
A 为 position: absolute 的情况



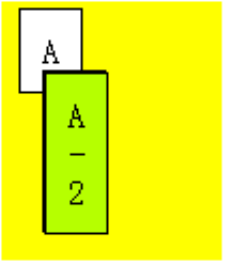
A - 1 为 position: absolute 的情况



A 和 A-1 均为 position: relative 的情况



A 和 A-1 均为 position: absolute 的情况

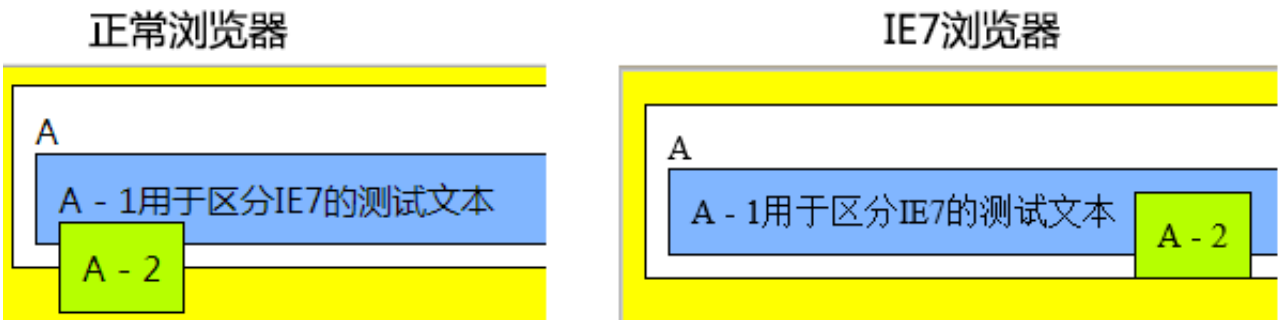


从上面的图，我们可以总结以下几个结论。

1）块状元素在position(relative/static)的情况下width为100%，但是设置了position: absolute之后，会将width变成auto（会受到父元素的宽度影响）。

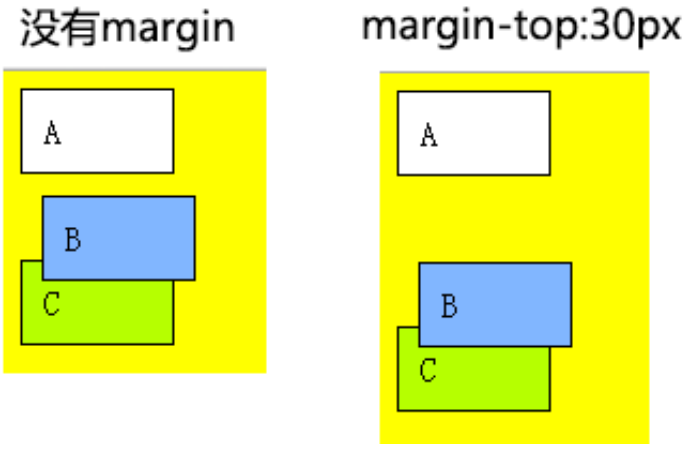
2）元素设置了position: absolute之后，如果没有设置top、bottom、left、right属性的话，浏览器会默认设置成auto，而auto的值则是该元素的“默认位置”。即设置position: absolute前后的offsetTop和offsetLeft属性值不变。
特殊情况：

- Firefox的话会直接将top、left设置成offsetTop和offsetLeft的值而非auto。
- IE7下的表现更类似于float，会附加到父元素的末尾。



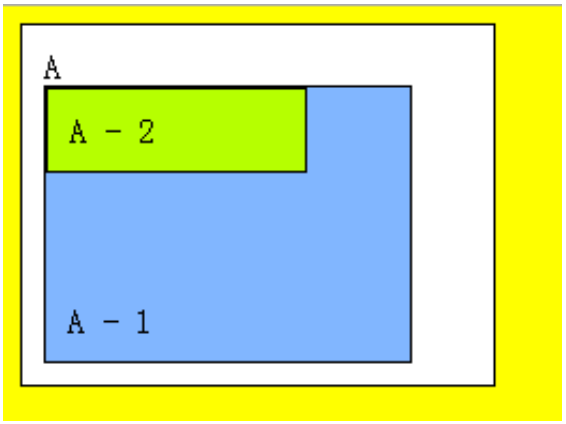
一些的position小知识

1）应用了position: relative/absolute的元素，margin属性仍然有效，以position:relative来举例。如果设置了left、top、bottom、right的属性，建议大家不要设置margin数据，因为很难精确元素的定位，尽量减少干扰因素。



2）position: absolute忽略根元素的padding。

```
<div id="a" style="background: #fff; width: 200px;">A
    <div id="b" style="background: #81b6ff; width: 150px;
position: relative; padding-top: 100px;">A - 1
        <div id="c" style="background: #b6ff00; position:
absolute; left: 0; top: 0">A - 2
        </div>
    </div>
</div>
```

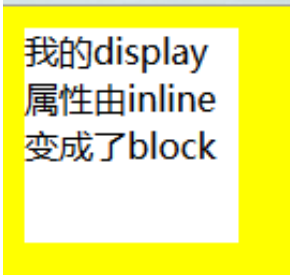


3）在IE6/7中设置position属性后会导致z-index属性失效

```
<!-- 解决方案，父元素设置一个更大的z-index值即可 -->
<div style="z-index: 2;">
  a
    <div style="position: relative; z-index: 1;">
      b
    </div>
</div>
```

4）行内元素在应用了position：absolute之后会改变display。

```
<span style="position: absolute; width: 100px; height:
100px; top: 10px; left: 10px; background: #fff;">
        我的display属性由inline变成了block
</span>
```



因此，要注意到relative是并没有改变行内元素的呈现模式，而absolute是会改变行内元素的呈现模式，如果设置了absolute并不需要显式的的将元素display改成block。

5）应用了position: absolute / relative之后，会覆盖其他非定位元素（即position为static的元素），如果你不想覆盖到其他元素，也可以将z-index设置成-1。

Fixed

在很长的时间里，这个属性值因为兼容性问题，并没有得到非常广泛的应用（IE6未实现该属性值）。fixed和absolute有很多共同点：

1. 会改变行内元素的呈现模式，使display之变更为block。
2. 会让元素脱离普通流，不占据空间。
3. 默认会覆盖到非定位元素上。

fixed与absolute最大的区别在于：absolute的”根元素“是可以被设置的，而fixed则其”根元素“固定为浏览器窗口。即当你滚动网页，其元素与浏览器窗口之间的距离是恒定不变的。

Float属性

float的属性值有none、left、right，有几个要点：

1. 只有横向浮动，并没有纵向浮动。

2. 当元素应用了float属性后，将会脱离普通流，其容器（父）元素将得不到脱离普通流的子元素高度。

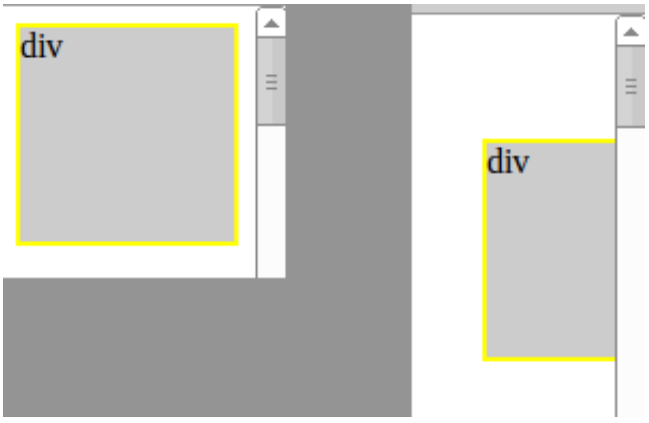
3. 会将元素的display属性变更为block。

4. 浮动元素的后一个元素会围绕着浮动元素（典型运用是文字围绕图片），与应用了position的元素相比浮动元素并不会遮盖后一个元素。
5. 浮动元素的前一个元素不会受到任何影响（如果你想让两个块状元素并排显示，必须让两个块状元素都应用float）。

与position的兼容性问题

1）元素同时应用了position: relative、float、（top / left / bottom / right）属性后，则元素先浮动到相应的位置，然后再根据（top / left / bottom / right）所设置的距离来发生偏移。

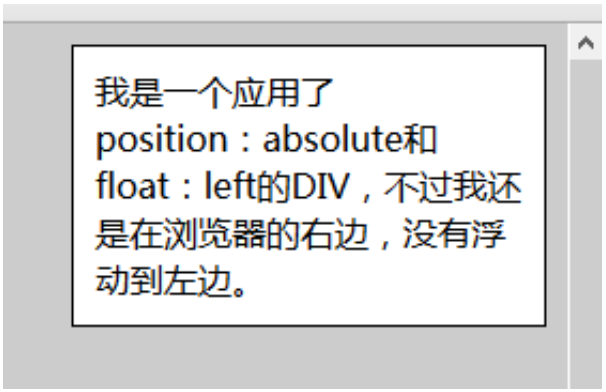
```
<div style="position:relative; float:right; left:50px;
top:10px;">div</div>
```



左图中的div是没有设置top、left值的，而右边则设置了50px的偏移。

2）元素同时应用了position: absolute及float属性，则float失效。

```
<div style="position: absolute; right:10px; top: 10px;
float: left;">
我是一个应用了position：absolute和float：left的DIV，不过我还是在浏览器的右边，没有浮动到左边。
</div>
```



3）第一个元素应用了position之后会覆盖着接下来的float元素（如果两个元素所处的位置相同）

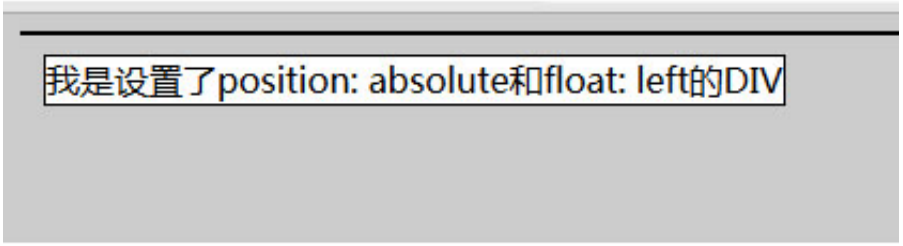
```
<div style="position: absolute; left:10px; top: 10px;">
    我是一个应用了position：absolute的DIV。
</div>
<div style="float:left; background: red; width: 300px;
height: 150px; ">
    我是float：left的DIV
</div>
```

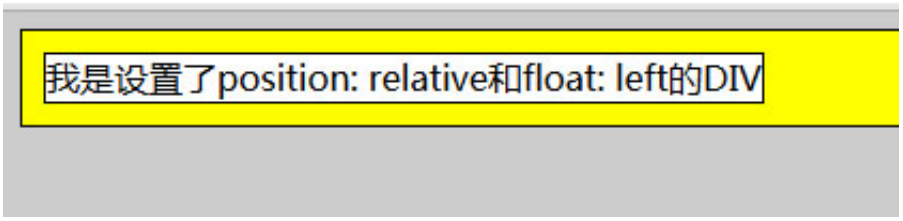
> 回顾：如果你不将float的元素的position设置成relative的话，你想通过设置float元素的z-index来的达到覆盖position:absolute是无效的。同理，float元素下面存在position: absolute的子元素，如果你不将float的元素的position设置成relative的话，absolute元素是不会定位到float元素的。

4）同时应用position: absolute和float: left会导致清除浮动无效（position: relative则可以清除浮动）。
常用的清除浮动的方法有两种：

1. 通过在容器中添加一个标签，设置该标签的样式为 clear: both
2. 容器设置overflow: hidden

```
<div style="background: #fff; width: 100%; overflow:
hidden;">
    <div style="float: left; position: absolute;">我是
DIV</div>
    <div style="clear: both;"></div>
<div>
```





最后，如果你觉得这篇文章对你有用的话。请帮忙点一下推荐，谢谢！^_^

1. 在睡觉前一个小时洗一个热水澡，然后等身体逐渐变凉。有助于快速入眠
2. 睡眠限制计划。每天现在在卧室的时间如：6个小时，其他任何时间都不能够进入卧室或者睡觉。每天都在固定的时间起床如：早上8点。
3. 打盹法。每次打盹大约30分钟。早上7点-12点之间及晚上6点-8点之间不能打盹，最佳的时间为下午2-5点。
4. 打鼻鼾，可以通过一些非处方药或者设备来减缓症状。
5. 咖啡和酒。睡眠分为5个阶段。1）昏昏欲睡 2）轻度睡眠 3-4）深度睡眠 5）做梦（眼睛快速转动）。每个阶段大约持续1.5小时，健康的睡眠会经历4-6个阶段。咖啡会导致进入睡眠的时间增加，轻度睡眠的时间增加而深度睡眠的时间减少。酒会快速入睡，当时进入眼睛快速转动的时间会增加，会导致下半夜醒来的次数增加。
6. 睡醒的时候通过模拟日光的灯照射能够使人快速清醒（蓝色的灯光），场景是凌晨需要起来。
7. 高蛋白质的食物能够使人清醒而碳水化合物则有助于睡眠。晚上睡觉前四个小时就要吃晚餐。
8. 通过长时间禁食能够调整生物钟来适应时差。
9. 肌肉紧张放松物理法，能够帮助进入睡眠。控制身体每个部分绷紧然后放松，遍布全身。
10. 薰衣草泡茶喝有助于睡眠。