

# Wayne Zhu

Keep looking, don't settle.

博客园

首页

新随笔

联系

管理

随笔 - 525 文章 - 0 评论 - 254

## 深入理解css中position属性及z-index属性

## 深入理解css中position属性及z-index属性

在网页设计中，position属性的使用是非常重要的。有时如果不能认识清楚这个属性，将会给我们带来很多意想不到的困难。

position属性共有四种不同的定位方法，分别是static、fixed、relative、absolute,sticky。最后将会介绍和position属性密切相关的z-index属性。

### 第一部分：position: static

**static定位是HTML元素的默认值，即没有定位，元素出现在正常的流中，因此，这种定位就不会收到top, bottom,left,right的影响。**

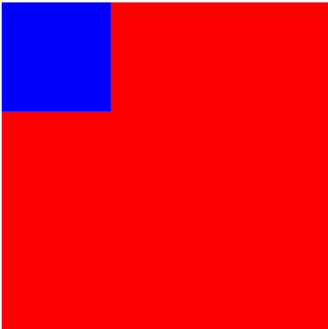
如html代码如下：

```
1 <div class="wrap">
2   <div class="content"></div>
3 </div>
```

css代码如下：

```
1 .wrap{width: 300px;height: 300px; background: red;}
2 .content{position: static; top:100px; width: 100px;height: 100px; background: blue;}
```

效果图如下：



我们发现，虽然设置了static以及top，但是元素仍然出现在正常的流中。

### 第二部分：fixed定位

**fixed定位是指元素的位置相对于浏览器窗口是固定位置，即使窗口是滚动的它也不会滚动，且fixed定位使元素的位置与文档流无关，因此不占据空间，且它和其他元素发生重叠。**

html代码如下：

```
1 <div class="content">我是使用fix来定位的!!!所以我相对于浏览器窗口，一直不动。</div>
```

css代码如下：

```
1 body{height:1500px; background: green; font-size: 30px; color:white;}
2 .content{ position: fixed; right:0;bottom: 0; width: 300px;height: 300px; background: bl
```

搜索

找找看

谷歌搜索

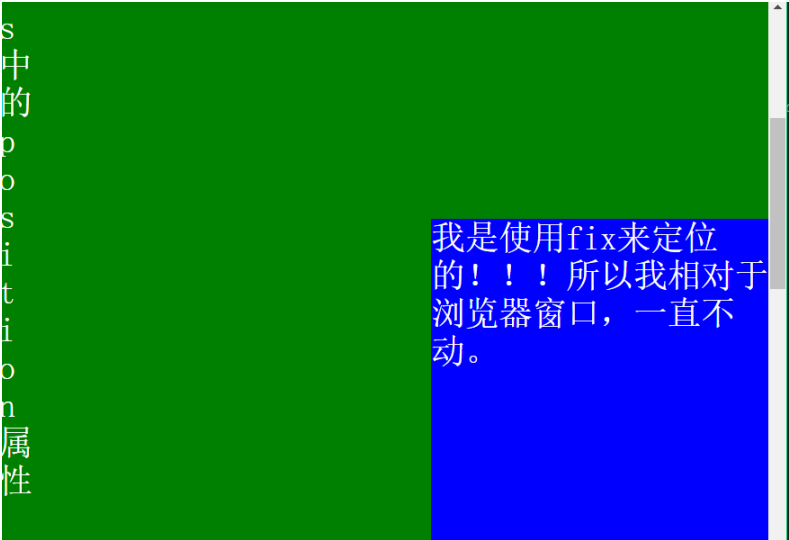
积分与排名

积分 - 565925  
排名 - 464

随笔分类

- Ajax(3)
- Angular(1)
- Bootstrap(1)
- C/C++(30)
- cars(2)
- chrome(7)
- computer problems && solutions (5)
- CSS3(11)
- DOS(3)
- Electron(1)
- ES6(7)
- flow.js / typescript(2)
- fortran(1)
- git&&svn(6)
- HTML&CSS(48)
- HTML5(14)
- java(7)
- JavaScript(83)
- jQuery(5)
- JSON&&XML(4)
- Machine Learning(3)
- mgp(8)
- node(22)
- photoshop与切图(3)
- PHP(2)
- python(4)
- React(22)
- Regular Expression(4)
- sass/less(4)
- vue(38)
- web安全(2)
- wechat(4)
- 产品(10)
- 常用软件和插件(10)
- 关于前端(19)
- 计算机(12)
- 计算机专业基础(24)
- 可视化(11)
- 前端工程化(11)
- 前端性能优化/效率(17)
- 前后端分离(3)
- 实例(4)
- 数据库(8)
- 网站记录(4)
- 问题记录(16)
- 移动端(11)

效果图如下：



即右下角的div永远不会动，就像经常弹出来的广告!!!

值得注意的是：fixed定位在IE7和IE8下需要描述！DOCTYPE才能支持。

### 第三部分：relative定位

相对定位元素的定位是相对它自己的正常位置的定位。

关键：如何理解其自身的坐标呢？

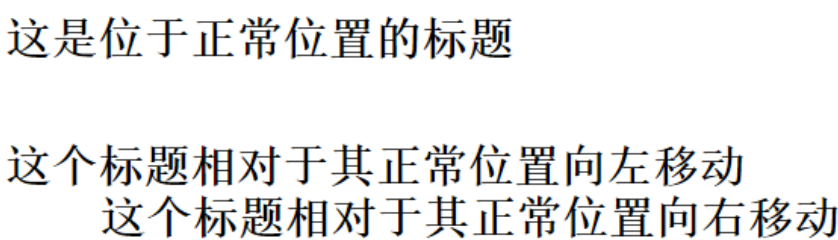
让我们看这样一个例子，html如下：

```
1 <h2>这是位于正常位置的标题</h2>
2 <h2 class="pos_bottom">这个标题相对于其正常位置向下移动</h2>
3 <h2 class="pos_right">这个标题相对于其正常位置向右移动</h2>
```

css代码如下：

```
1 .pos_bottom{position:relative; bottom:-20px;}
2 .pos_right{position:relative;left:50px;}
```

效果图如下：



即bottom:-20px;; 向下移动。 left:50px;向右移动。

即可以理解为：移动后是移动前的负的位置。

比如上例中，移动后是移动前负的bottom:-20px;即移动后是移动前bottom:20px;也就是说，移动后是移动前的向下20px；

又如：left:50px;移动后是移动前左边的-50px;那么也就是说移动后是移动前的右边的50px。

即：移动后对于移动前：如果值为负数，则直接换成整数；如果值为整数，则直接改变相对方向。

用户体验(8)  
杂文(47)  
转载文章(12)

随笔档案

- 2019年9月(1)
- 2019年1月(1)
- 2018年12月(3)
- 2018年11月(1)
- 2018年10月(2)
- 2018年7月(3)
- 2018年6月(6)
- 2018年5月(13)
- 2018年4月(25)
- 2018年3月(19)
- 2017年12月(5)
- 2017年11月(14)
- 2017年10月(11)
- 2017年9月(16)
- 2017年8月(35)
- 2017年7月(38)
- 2017年6月(32)
- 2017年5月(76)
- 2017年4月(35)
- 2017年3月(32)
- 2017年2月(48)
- 2017年1月(15)
- 2016年12月(43)
- 2016年11月(37)
- 2016年10月(14)

相册

photos(1)

最新评论

- 1. Re:windows下使用curl命令 && 常用curl命令 win10 内嵌了 --javakam
- 2. Re:原生js实现轮播图 没有初始化wrap的值可能导致无法实现轮播。 --Xerox!
- 3. Re:原生js实现轮播图 跟着楼主学所有功能全部实现 暂时没有发现任何bug --我爱吃番茄鸡蛋
- 4. Re:图片懒加载原理及实现 Nice --青栀i
- 5. Re:状态码301和302的区别 301重定向的例子有错误，百度使用的临时重定向302 --realzhaijiayu

阅读排行榜

弄清楚了relative是如何移动的，下面我们看一看移动之后是否会产生其他的影响。

html代码如下：

```
1 <h2>这是一个没有定位的标题</h2>
2 <h2 class="pos_top">这个标题是根据其正常位置向上移动</h2>
3 <p><b>注意:</b> 即使相对定位元素的内容是移动,预留空间的元素仍保存在正常流动。</p>
```

css代码如下：

```
1 h2.pos_top{position:relative;top:-35px;}
```

效果图如下：

这个标题是根据其正常位置向上移动

注意：即使相对定位元素的内容是移动,预留空间的元素仍保存在正常流动。

根据之前的说法，top:-35px; 值是负数，则直接换成正数，即移动后相对与移动前向上偏移了35px；我们发现于上，移动后和上面的元素发生了重叠；于下，即使相对元素的内容移动了，但是预留空间的元素仍然保存在正常流动，也就是说相对移动之后，不会对下面的其他元素造成影响。

注意：top：20px；是指子元素的margin外侧和包裹元素的border内侧之间的距离是20px。

第四部分：absolute定位

绝对定位的元素相对于最近的已定位父元素，如果元素没有已定位的父元素，那么它的位置相对于<html>。

下面举几个例子：

例子1：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>绝对定位</title>
6   <style>
7     body{background:green;}
8     .parent{ width: 500px;height: 500px;background: #ccc;}
9     .son{ width: 300px;height: 300px;background: #aaa;}
10    span{position: absolute; right: 30px; background: #888;}
11  </style>
12 </head>
13 <body>
14   <div class="parent">
15     <div class="son">
16       <span>什么? </span>
17     </div>
18   </div>
19 </body>
</html>
```

效果如下：

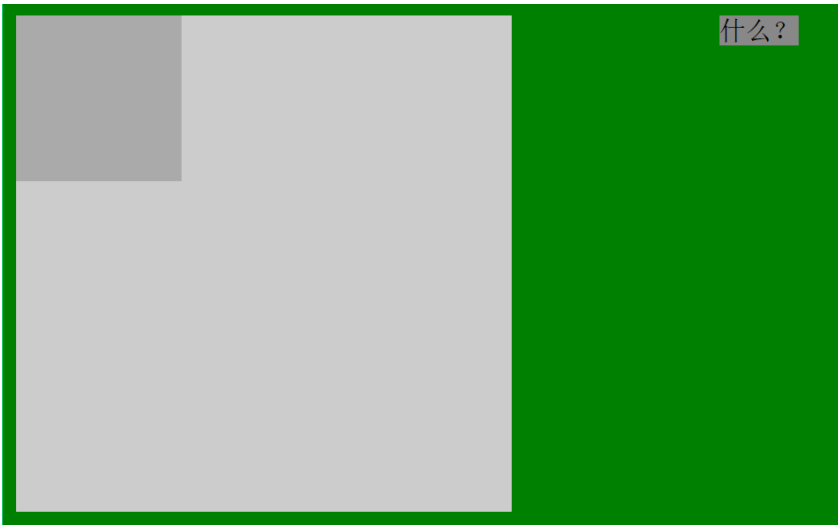
- 1. 原生js实现轮播图(97086)
- 2. 状态码301和302的区别(87742)
- 3. vscode下调试运行c++(84693)
- 4. 正则表达式之数字验证(75638)
- 5. windows下使用curl命令 && 常用curl命令(58623)

评论排行榜

- 1. 原生js实现轮播图(27)
- 2. jQuery中的100个技巧（译）(27)
- 3. 总结(20)
- 4. 深入理解css中的margin属性(13)
- 5. 深入理解css中position属性及z-index属性(11)

推荐排行榜

- 1. jQuery中的100个技巧（译）(28)
- 2. 原生js实现轮播图(25)
- 3. 深入理解css中position属性及z-index属性(22)
- 4. 总结(20)
- 5. 如何写出优雅的css代码？(13)

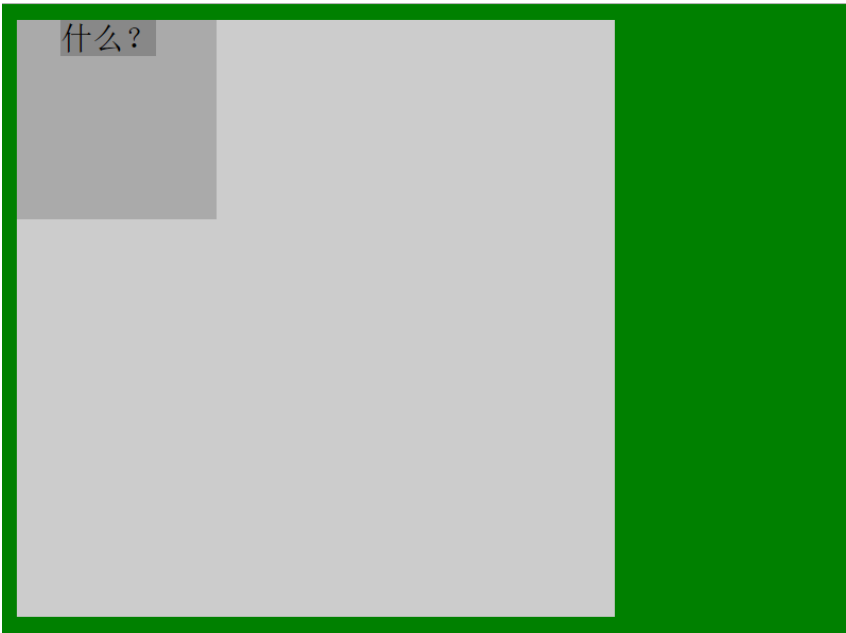


即我只在span中设置了position:absolute; 而在其父元素中都没有, 于是它的位置是相对于html的。

例2:

```
1 | .son{position: relative; width: 100px;height: 100px;background: #aaa; }
```

相较于上一个例子, 我只修改了class为son的元素的css, 设置为position: relative; 效果图如下:

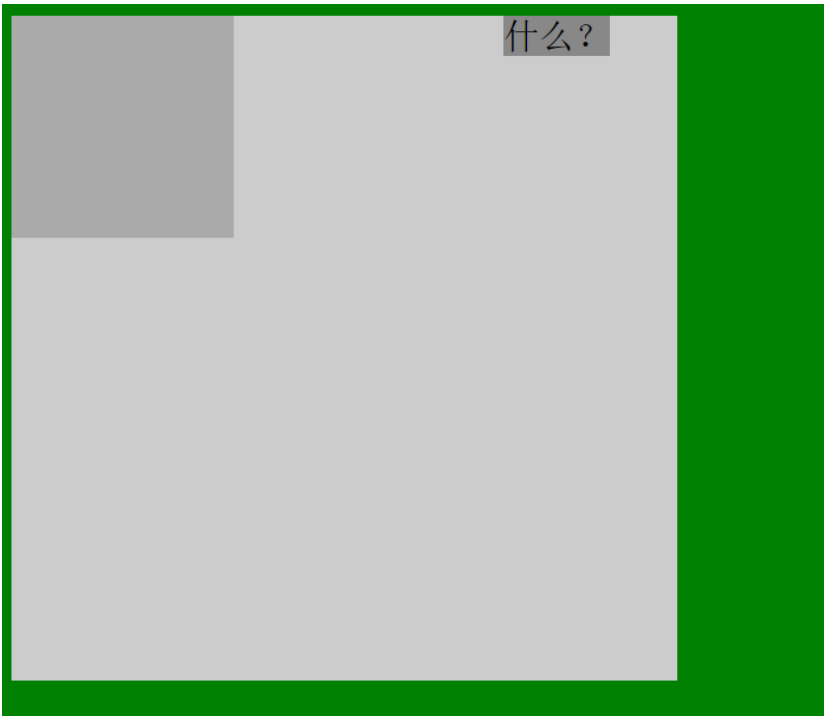


于是, 我们发现现在span的位置是相对于设有position属性的class为son的父元素的。

例3:

```
1 | .parent{position: absolute; width: 300px;height: 300px;background: #ccc;}
```

这个例子我只是修改了**第一个例子**中的css--设置了position:absolute; 效果如下:



于是我们发现，现在span的定位是相对于具有position:absolute的属性的class为parent的父元素。

例4：

```
1 | .parent{position:fixed; width: 300px;height: 300px;background: #ccc;}
```

相对于例1，我添加了fixed的position属性，发现结果和例3是一模一样的。

例5：

```
1 | .parent{position:static; width: 300px;height: 300px;background: #ccc;}
```

相对于例1，我添加了static的position属性（即html的默认属性），结果和例1是一样的。

综上所述，当某个absolute定位元素的父元素具有position:relative/absolute/fixed时，定位元素都会依据父元素而定位，而父元素没有设置position属性或者设置了默认属性，那么定位属性会依据html元素来定位。

## 第五部分：重叠的元素--z-index属性

首先声明：z-index只能在position属性值为relative或absolute或fixed的元素上有效。

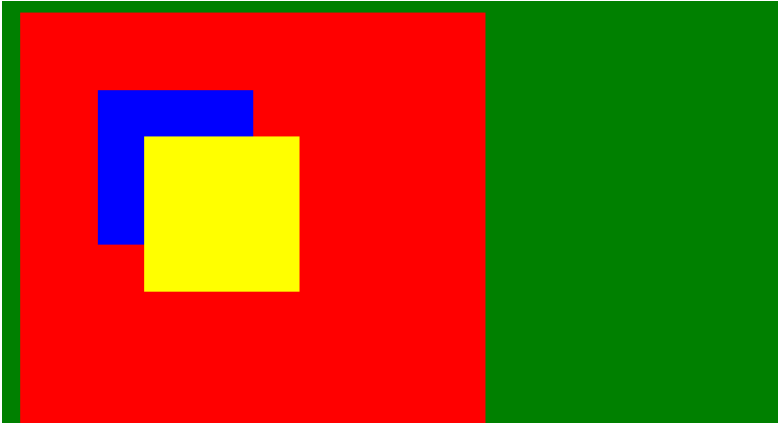
基本原理是：z-index的值可以控制定位元素在垂直于显示屏幕方向（z轴）上的堆叠顺序(stack order),值大的元素发生重叠时会在值小的元素上面。

下面我们通过几个例子继续来理解这个属性。

例1：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>绝对定位</title>
  <style>
    body{background: green;}
    .parent{position: relative; right: -5px; width: 300px;height: 300px;background: red;}
    .son1{position: absolute; left: 50px;top:50px; width: 100px;height: 100px;background: blue; }
    .son2{position: absolute; left: 80px;top: 80px; width: 100px;height: 100px;background: yellow; }
  </style>
</head>
<body>
  <div class="parent">
    <div class="son1"></div>
    <div class="son2"></div>
  </div>
</body>
</html>
```

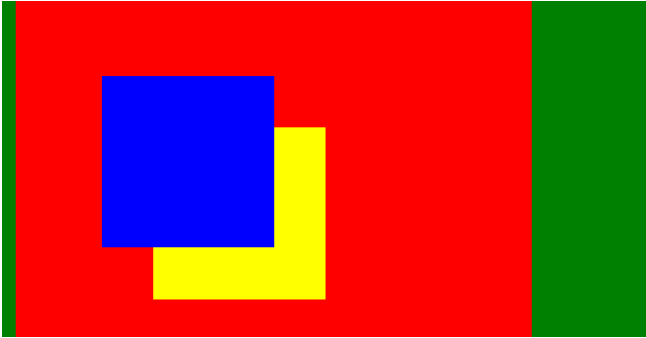
即son1和son2是parent的两个子元素，效果图如下：



这是没有使用z-index，我们发现son2在son1之上，这是因为son2在html中排在了son1之后，所以后来者将前者覆盖，如果我们颠倒以下两者的顺序，就会发现蓝色(son1)在上上了。

例2：

在son1中加入z-index: 1; 可以发现效果如下：

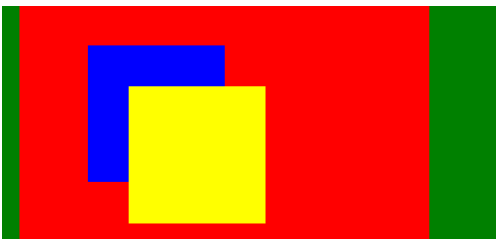


也就是说son2的index值是小于1的。

如果我们给son2也加上z-index:1;呢？结果得到黄色(son2)就在上面了。（因为一旦z-index值相等，情况就和都不设置index值一样了）

例3：

在son2中加入z-index:5;可以发现效果如下：



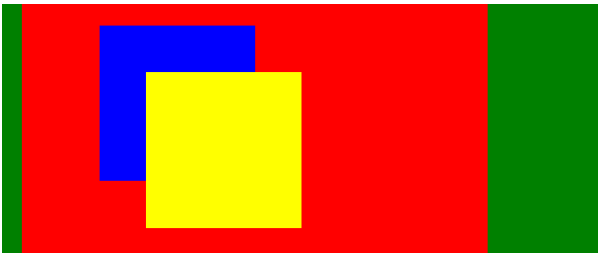
即黄色(son2)又在上面了，这个很简单，不作过多讨论。

例4：

在父元素添加z-index:10;

在son1和son2添加z-index:5; 这样理论上父元素就会在上面（黄色覆盖蓝色和黄色）；

结果如下：



结果没有变!!!!!! 这就说明父元素和子元素不能做z-index的比较!!! 但真的是这样吗? 看下一个例子:

例5:

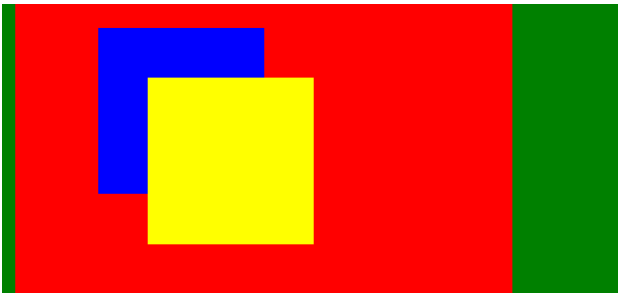
把两个子元素的z-index值同时设置为-5; 父元素不设置z-index属性。结果如下:



成功!! 说明在父元素和子元素之间还是可以作比较的!!! 只是需要我们把子元素的z-index值设为负数。

例6:

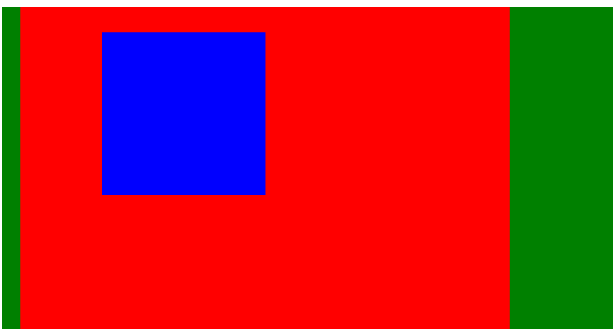
我们在例5的基础上再给父元素添加一个z-index:10, 讲道理~应该也可以得到和例5相同的结果吧!!



然而.... 看来我们不能设置父元素的z-index值, 否则就不能出现我们想要的效果。下面再看一个有趣的例子!

例7:

我们根据例6的经验不设置父元素的值, 现在设置son1(蓝色)的z-index为5, son2的z-index为-5, 看下面的结果:



即son1在最上面，父元素在中间，son2在最下面。

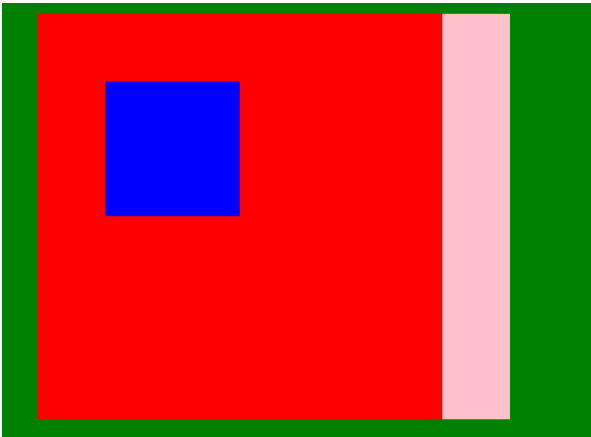
对于z-index的探索就这样结束了吗？？当然没有，看下面几个更为有趣的例子吧。

例8：

代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>绝对定位</title>
  <style>
    body{background: green;}
    .parent1{position: absolute; left: 50px; width: 300px; height: 300px; background: red; z-index: 15;}
    .parent2{ position: absolute; left: 100px; width: 300px; height: 300px; background: pink; z-index: 10;}
    .son1{position: absolute; left: 50px; top: 50px; width: 100px; height: 100px; background: blue; }
    .son2{position: absolute; left: 50px; top: 50px; width: 100px; height: 100px; background: yellow; }
  </style>
</head>
<body>
  <div class="parent1">
    <div class="son1"></div>
  </div>
  <div class="parent2">
    <div class="son2"></div>
  </div>
</body>
</html>
```

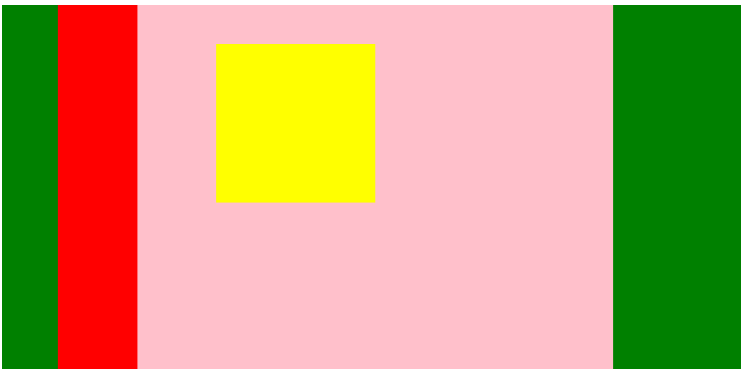
效果如下：



虽然parent1和parent2分别是son1和son2的父元素，按照我们之前的理解，父元素是不可添加z-index值的，否则会导致错误。但是这里parent1和parent2相对于body又是子元素，他俩是同级的，所以就可以进行比较了。且此时parent1的子元素son1（蓝色）在上。

例9：

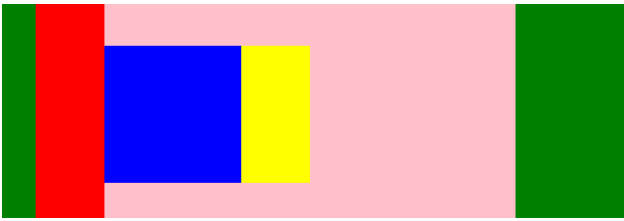
如果我们在例7的基础上，把parent2的z-index值设为20，就会发现如下效果：



即parent2在上的同时son2也会同时在上。这也就是所谓的“拼爹”了！！

例10.同样在例7的基础上，我们不设置parent1和parent2和son2的index值，而只设置son1的z-index值为10，效果如下：

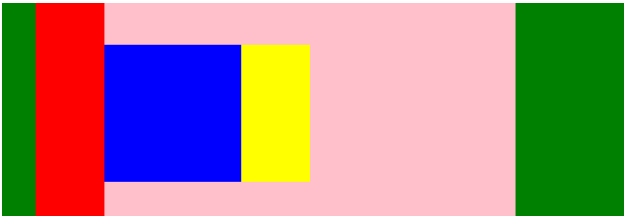




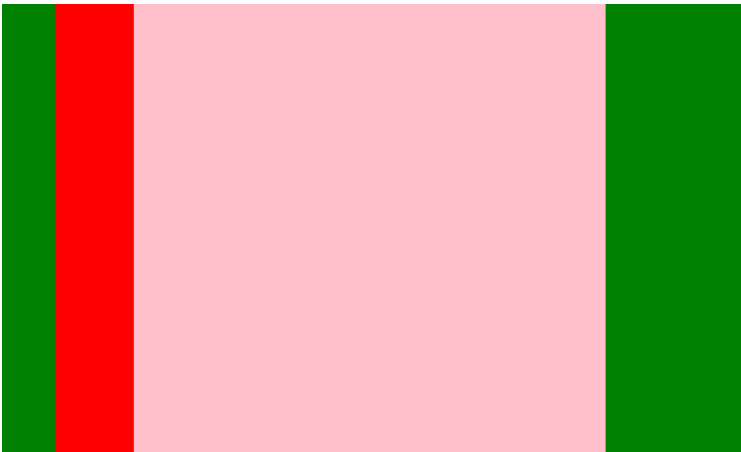
即原本在下面的蓝色son1被提上来了，而没有把父元素（parent1）提上了，诶，不孝顺啊！！

例11.显然，在例10的基础上，如果我们把son2的index值设置的比son1的大，如20，那么son2就会覆盖son1了，并且都在两个父元素只上了！！

效果如下图：



例12.当然，如果我们把两个son的z-index都设置位负数如-5，那么两者就都会被父元素所覆盖：



## 第六部分：脱离文档流导致的问题

我们知道如果使用position:absolute和position:fixed都会导致元素脱离文档流，由此就会产生相应的问题。举例如下：

```
<!DOCTYPE html>
<html>
<head>
  <title>position</title>
  <style>
    .div1{
      background-color: red;
      padding:20px;
    }
    .div2{
      width: 200px;
      height: 200px;
      background-color: blue;
    }
  </style>
</head>
<body>
  <div class="div1">
    <div class="div2"></div>
  </div>
```

```
</body>
</html>
```



这时效果如下：



即子元素将父元素撑了起来。

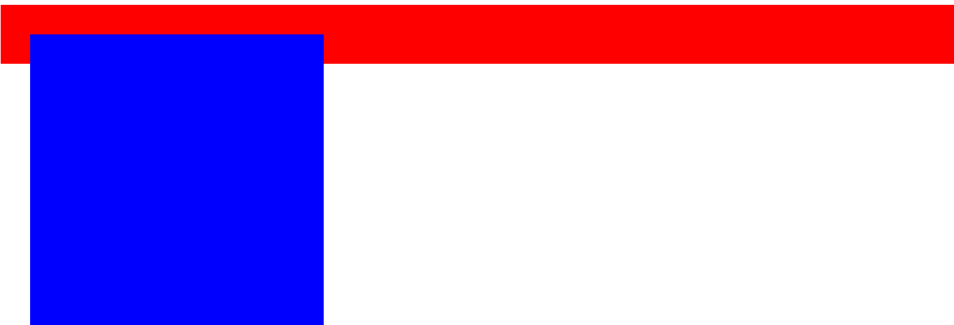
但是一旦子元素的`position`为`fixed`或者是`absolute`，那么它就会脱离文档流，这样的后果是父元素无法被撑开，如下所示：



```
<!DOCTYPE html>
<html>
<head>
  <title>position</title>
  <style>
    .div1{
      background-color: red;
      padding:20px;
      position: relative;
    }
    .div2{
      position: absolute; // 添加position:absolute使其脱离文档流
      width: 200px;
      height: 200px;
      background-color: blue;
    }
  </style>
</head>
<body>
  <div class="div1">
    <div class="div2"></div>
  </div>
</body>
</html>
```



最终效果如下所示：



解决方法1：在js中设置父元素高度等于子元素的高度。

解决方法2：给父元素强行设置高度。（对于宽度导致的类似问题就强行设置宽度）

这一部分内容是2017-3-13补充的，之前没有了解到这个属性，几天才知道这个属性就是我经常使用js使用的方法，就是在一个内容中，我们可以固定一个部分，然后到了另一个内容，又会固定另外一个部分。

[illegible]

[illegible]



这一部分知识还是非常有意思的，希望大家可以继续探索，当然如果通过这篇博文给大家一点点的帮助那就再好不过了！

<https://www.cnblogs.com/zhuozhenwei918/p/6112034.html>


好文要顶

关注我

收藏该文







Wayne-Zhu

关注 - 42

粉丝 - 337

+加关注

22

0

« 上一篇: [初识PHP](#)

» 下一篇: [计算机网络](#)

posted @ 2016-11-29 00:29 Wayne-Zhu 阅读(57361) 评论(11) 编辑 收藏

### 评论列表

#1楼 2017-09-18 19:59 gitlab_sam	回复 引用
简单易懂，支持楼主	支持(0) 反对(0)
#2楼 2017-10-28 00:32 临窗雨声	回复 引用
已收藏，谢过	支持(0) 反对(0)
#3楼 2018-02-27 10:33 tsailer	回复 引用
谢谢O(n_n)O哈哈~	支持(0) 反对(0)
#4楼 2018-04-04 10:36 _狼狼	回复 引用
十分感谢	支持(0) 反对(0)
#5楼 2018-04-13 15:49 IT小民工lxy	回复 引用
有用，简单易懂，谢谢啦!	支持(0) 反对(0)
#6楼 2018-11-04 13:56 你的鸭先生	回复 引用
demo图片挂了。。。	支持(0) 反对(0)
#7楼 2018-12-21 13:05 seasonwarm	回复 引用
相对于<html>	支持(0) 反对(0)
#8楼 2018-12-21 13:05 seasonwarm	回复 引用
你好，“相对于<html>” 不理解这个是什么意思? 和相对于<body>有什么区别?	支持(0) 反对(0)
#9楼 2019-01-10 15:25 皮实的柚子	回复 引用
支持	支持(0) 反对(0)
#10楼 2019-09-09 19:52 LIYIRONG	回复 引用

赞!

支持(0) 反对(0)

#11楼 2020-05-28 19:21 vampire-11

回复 引用





为了表达对您的感谢，我特意注册了用户。  
这篇文章帮我解决了一个困扰很久的问题，我从头看到尾，解释的很清楚。  
非常感谢。

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

发表评论

编辑 预览

B    

支持 Markdown

提交评论 退出 订阅评论

[Ctrl+Enter快捷键提交]

- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】1200件T恤+6万奖金，阿里云编程大赛报名开启
- 【推荐】未知数的距离，毫秒间的传递，声网与你实时互动
- 【推荐】了不起的开发者，挡不住的华为，园子里的品牌专区
- 【推荐】首次公开！三代技术人深度对话，《云上朗读者》开放下载

- 相关博文:
- 深入理解css中position属性及z-index属性
  - 理解css 中的position五个属性
  - CSS属性之position讲解
  - CSS position 属性
  - CSSposition(定位)属性
  - » 更多推荐...

- 最新 IT 新闻:
- 没有对手的滴滴 为什么还要烧钱做花小猪？
  - 那些走过的路、遇到的人、经历的事...都在塑造你的大脑
  - 他们为什么选择中科大少年班？官方公布48名新生名单

- 一块CPU就能运行超逼真水流特效，胡渊鸣的算法被这样实现，本人看了都说好
- 植入狂魔唯品会 为何只能兴盛在影视综艺里？
- » 更多新闻...

Copyright © 2020 Wayne-Zhu  
Powered by .NET Core on Kubernetes

一分耕耘，一分收获。