

# CSS 相对/绝对(relative/absolute)定位系列 (三)

这篇文章发布于 2011年03月20日，星期日，22:32，归类于 [css相关](#)。 阅读 93488 次, 今日 39 次

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址：<http://www.zhangxinxu.com/wordpress/?p=1528>

//zxx:直接接上回

## 一、absolute正业之元素隐藏

元素隐藏与显示是我们在页面制作与交互效果实现中非常常见的，如果您只是使用 `display:none` 与 `display:block/inline` 来实现DOM元素的显隐控制，那你就out了。就元素的显示与隐藏实现，使用 `display` 在有些时候算是比较糟糕的方法了。

控制元素显隐的方法很多，但是本文不是讲元素显隐控制的，所以，只讲与absolute相关的一些方法。

absolute属性相关的隐藏方法，我知道的有三种，分别如下：

```
.hidden{
    position:absolute;
    top:-9999em;
}
```

```
.hidden{
    position:absolute;
    visibility:hidden;
}
```

```
.hidden{
    position:absolute;
    clip: rect(1px 1px 1px 1px); /* IE6, IE7 */
    clip: rect(1px, 1px, 1px, 1px);
}
```

使用absolute属性控制DOM元素的显隐有三个关键点：**页面可用性**，**回流与渲染**，**配合JavaScript的控制**。

### ① 可用性隐藏

所谓可用性隐藏，就是兼顾屏幕阅读器这类互联网阅读辅助设备的隐藏方式。Yahoo! 可用性实验室成员Ted Drake<sup>[1]</sup>就不同隐藏方法下屏幕阅读器的可用性问题作为测试，结果发现下面两种隐藏方式屏幕阅读器是读不了的。

```
.completelyhidden {
    display:none;
}
```

```
.visibilityhidden {
  visibility:hidden;
}
```

You don't want to show those hidden panels to any user. Use **display:none** for the hidden panels.

Screen readers will also ignore sections with **visibility:hidden**.

所以，从可用性角度而言，像“选项卡内容”，“更多收起展开”这类元素隐藏与显示就不推荐使用 `display:none`，或者是 `position:absolute + visibility:hidden`。

例如优酷网电影或视频的简介中“显示详情”的实现就是使用的 `display:none`，如下截图：

优酷网的展开详情display:none实现 张鑫旭-鑫空间-鑫生活

而大众点评网的隐藏层多采用 `position:absolute + visibility:hidden` 的方法，如下截图：

大众点评网隐藏层的实现 张鑫旭-鑫空间-鑫生活

上述隐藏内容其实都是有用的信息，对于像盲人这类需要借助屏幕阅读器的用户无法知道这些信息了。拿优酷的那个例子，盲人用户就无法知道影片完整的简介。

如果希望隐藏内容能够被辅助阅读设备识别，就不能使用 `display:none` 以及 `visibility:hidden` 隐藏元素。可以使用模拟隐藏的隐藏方法，又称可用性隐藏。就是下面两种隐藏方法。

```
.hidden{
  position:absolute;
  top:-9999em;
}
```

```
.hidden{
  position:absolute;
  clip: rect(1px 1px 1px 1px); /* IE6, IE7 */
  clip: rect(1px, 1px, 1px, 1px);
}
```

但是，如果你是希望完全隐藏的，那就可以使用 `display:none` 或 `visibility:hidden`。

**额外说明：**如果隐藏元素含有链接元素或是可获得焦点的控件元素，但是又是使用的可用性隐藏。这些隐藏的连接与控件也是可以响应键盘焦点Tab切换的，但是这会让键盘使用用户产生不解与疑惑的。所以，从某种意义说，某些情况下，要兼顾屏幕阅读器用户和键盘用户有时候是不可兼得的。

## ②回流与渲染

早先时候我曾翻译过两篇关于回流与重绘的文章，“[最小化浏览器中的回流\(reflow\)](#)”以及“[回流与重绘：CSS性能让JavaScript变慢？](#)”。

我自己是没测过。不过根据上面这两篇文章的说法，以及一位口碑前端前辈的说法，使用absolute隐藏于显示元素是会产生重绘而不会产生强烈的回流。而使用 `display:none` 不仅会重绘，还会产生回流，DOM影响范围越广，回流越强烈。所以，就JavaScript交互的呈现性能上来讲，使用absolute隐藏是要优于display相关隐藏的。

### ③配合JavaScript的控制

说到元素的显示与隐藏，免不了与JavaScript的交互。例如display相关的隐藏于显示，就是 `display:block/inline/inline-block/...` 与 `display:none`。

要让元素隐藏，很简单，直接：

```
dom.style.display = "none";
```

但是，如果要显示隐藏的元素，咋办呢？因为不同的标签所处的display水平是不一样的，于是，我们很难有一个简单的统一的显示方法。例如，下面的代码可能使用于 `div`，`p` 标签，但是对于span等inline水平的元素，可能就会囧屁了（原本单行显示结果换行）。

```
dom.style.display = "block";
```

况且，随着浏览器的不断进步，以后类似于 `display:table-cell`，`display:list-item` 会越来越多的使用。再想通过display实现通用的显隐方法难度又会增大些。

这就是使用display属性控制元素显隐的局限性。顺带一提的是jQuery的显隐方法 `show()/hide()/toggle()` 就是基于 `display` 的，其会存储元素先前的 `display` 属性值，于是元素再显示的时候就可以准确地显示出之前的 `display` 值了。

您可以狠狠地点击这里：[jQuery与display的显隐测试](#)。

jQuery存储display数组值 张鑫旭-鑫空间-鑫生活

而使用绝对定位实现的一些元素隐藏方法的控制就相对简单很多的。例如：`position:absolute + visibility:hidden` 方法，当我们要让元素（原本非绝对定位元素）显示的时候，我们需要设置：

```
dom.style.position = "static";  
dom.style.visibility = "visible";
```

而类似的 `position:absolute + top:-999em` 方法，当我们要让元素（原本非绝对定位元素）显示的时候，我们只需要设置：

```
dom.style.position = "static";
```

而无需担心原本标签的是inline水平还是block水平。所以，就显隐的JavaScript控制上来讲，absolute相关方法要比display略胜一筹。

结合上面三点讨论，我们可以看出，当前占据主流的 `display:block/none` 控制元素显示与隐藏的方法其实是诸多方面有弊端的方法，有拿着鸡毛当令箭的意味。实际上，这种活（元素显隐）交给absolute属性更合适，**控制元素显示与隐藏才是absolute属性的正业所在**。//zxx: display属性控制元素显隐之所以会控制大半壁江山是因为其语义就是“显示(display)”，于是先入为主，再加上人的从众性。

## 二、absolute与等高布局

拿简单的两栏布局举例，左栏与右栏有不同的背景色，且中间隔边框线分隔，如何实现？因为随着内容的不同，有可能左侧栏高度较高，也有可能是右侧栏高度较高。所以，要实现无缝的填色，定高不

行不通的，置高度不理显然也不行，此时解决方法就是让左右两栏等高。

我较早的时候写过一篇名为“[纯CSS实现侧边栏/分栏高度自动相等](#)”的小tip，其实现原理如下：

```
margin-bottom:-3000px; padding-bottom:3000px;
```

后来在“[我所知道的几种display:table-cell的应用](#)”一文中也提过使用 `display:table-cell` 实现等高布局。

这里再介绍些如何使用 `absolute` 实现等高布局。

正如系列[前篇](#)所述，应用了 `position:absolute` 的元素无宽度，无高度。正好，我们可以利用该特性来实现等高布局所需要的效果——如等高的背景色、边框效果等。

您可以狠狠地点击这里：[绝对定位与等高布局demo](#)。

点击demo页面中的两个按钮就可以看到无论左侧栏高还是右侧栏高，两边背景颜色纯纯的，中间的垂直分隔线直直的，如下截图：

绝对定位等高布局左侧栏高度较高 张鑫旭-鑫空间-鑫生活

绝对定位等高布局右侧栏高度高 张鑫旭-鑫空间-鑫生活

其中，实现等高效果的核心CSS代码如下：

```
.equal_height{width:100%; height:999em; position:absolute; left:0; top:0;}
```

同时，满足以下一些条件：

1. 高度999em的绝对定位层位于侧栏容器内，侧栏 `position` 为 `relative`
2. 该栏实际元素内容用一个与absolute绝对定位层为兄弟关系的标签层包裹，`position` 为 `relative`，`z-index` 值 1 或其他
3. 左右栏的父标签需设置 `overflow:hidden`，同时为了兼容IE6/7，需设置 `position` 为 `relative`

以上条件对应下图标注：

绝对定位等高布局实现条件标注 张鑫旭-鑫空间-鑫生活

原理很简单：由于绝对定位元素无高度的特性无宽度的特性，我们可以伪造一个高度足够高的绝对定位层（设置背景色，边框等属性），同时设置父标签溢出隐藏，那么其多出来的高度就不会显示了，也就实现了看上去的等高布局效果了。具体细节可参见[demo页面](#)中的代码展示，相信很好理解的。

### 三、absolute属性与IE6/IE7之间的误会

`absolute`属性确实存在不少兼容性的问题，首先`absolute`属性定位相关(left/top)的些bug（例如IE6的奇偶bug）这里不予以讨论。//zxx:很多人都知道，再说就没意思了。

所以，下面所展示的些“误会”都是没有定位属性的(即无left/top/right/bottom)。

## 1. margin定位元素绝对定位元素重叠的误会

以前经常碰到的，今天怎么都模拟不出来了，这个先空着，回头补上…… 😊 (\*^\_\_^\*) 嘻嘻……

**补充于2011-04-18**

很简单，双栏自适应布局中，左侧元素absolute绝对定位，右侧的margin撑开距离定位。可参见“[页面重构鑫三无准则 之无宽度准则](#)”一文中新浪微博的实例页面中的使用（那里是padding撑开距离）。

为了再现IE6/IE7下的这个argin定位元素绝对定位元素重叠的误会，我特地做了个简单的demo页面，您可以狠狠的点击这里：[IE6/7下margin与absolute元素重叠demo](#)。

可以看到在IE6/7下左侧应用了absolute属性的图片与右边的自适应的文字内容重叠了，如下图所示（截自IE7）：

IE7下margin元素与absolute元素重叠 张鑫旭-鑫空间-鑫生活

此问题出现的原因与下面浮动与绝对定位元素重叠有着某些类似的原因，因为问题的出现都与绝对定位元素所在的标签水平有关：上述demo中，absolute属性所在的标签是div标签，属于block水平的元素。如下截图：

使用的div标签 张鑫旭-鑫空间-鑫生活

要是我们把这里的block水平的div元素修改成inline水平的span标签，则重叠的问题就没有了。如下截图：

span标签下的重叠问题的修复 张鑫旭-鑫空间-鑫生活

您可以狠狠的点击这里：[IE6/7下margin与absolute元素重叠问题修复demo](#)。

如果您手上的浏览器是即使是IE6/IE7，点击上面的demo页面也不会有重叠的bug了，而此问题的修复是非常简单的将div标签换成span，如下截图所示：

换成span标签的示意截图 张鑫旭-鑫空间-鑫生活

## 2. 浮动与绝对定位元素重叠的误会

很简单，前面一个标签是浮动元素，后面的是block水平的绝对定位元素，结果IE8+，以及现代浏览器文字与图片重叠；但是IE6/IE7浏览器确是并排显示的。如下截图示意：

IE8下文字与图片重叠 张鑫旭-鑫空间-鑫生活

IE6下图文并排 张鑫旭-鑫空间-鑫生活

您可以狠狠的点击这里：[浮动元素绝对定位元素重叠demo](#)。

CSS代码如下：

```
.box{padding:1em; background-color:#f0f3f9; overflow:hidden; _zoom:1;}
.l{float:left;}
.abs{position:absolute;}
```

HTML代码如下：

```
<div class="box">
  
  <div class="abs">呦喝，这不是张含韵小姐吗？</div>
</div>
```

为何现代浏览器以及IE8+浏览器下浮动的图片与绝对定位的文字会重叠，而IE6/7却是并列显示？这是由于IE6/IE7浏览器将inline水平标签元素和block水平的标签元素没有加以区分一视同仁渲染了。我在前面已经多次提到，**应用了绝对定位属性的元素具有包裹性，等同于没有高度与宽度的inline-block元素。**

上面斜体加粗的这个结论实际上说得不够严谨，在IE6/IE7浏览器下，上面的话是没错的；在所有浏览器下，对于inline水平的元素，上面的话也是没错的；但是在现代浏览器下，对于block水平的元素，上面的结论就有商榷之处。实际上，按照正确的绝对定位渲染，像 `div`，`p` 这类block水平标签并未完全inline-block化。inline-block化的元素有三大特性：包裹性；高宽可定义；图文混排。然而，`div`，`p` 这类标签应用了 `position:absolute` 后，在非IE6/7浏览器下，只有包裹性和高宽可定义这两个特性，但并不支持图片混排，也就是与图片文字在一起的时候会换行。

下面是举例字，验证上面的结论。首先是这么句话：“对于inline水平的元素，上面的话也是没错的”。这句话的意思其实是，如果是inline水平的元素，上面的那个示例就不会有兼容性问题了，于是我们把应用了 `abs` 类名的 `div` 标签改成 `span`，如下HTML代码：

```
<div class="box">
  
  <span class="abs">呦喝，这不是张含韵小姐吗？</span>
</div>
```

结果如下图：

IE8下浮动元素与绝对定位元素并排显示 张鑫旭-鑫空间-鑫生活

不仅IE8浏览器，Firefox/Chrome等先前重叠的现在都并排显示了。

您可以狠狠地点击这里：[浮动与inline水平绝对定位元素不重叠demo](#)

下面再来验证这个结论：“现代浏览器下block水平元素absolute化后不支持图片混排”。也是很简单的，我们可以把最上面重叠的那个例子的图片的浮动属性干掉，也就是如下的HTML代码：

```
<div class="box">
  
  <div class="abs">呦喝，这不是张含韵小姐吗？</div>
</div>
```

结果如下图：

IE8下绝对定位元素与图片换行 张鑫旭-鑫空间-鑫生活

而在IE6/IE7浏览器下，依旧是并排显示滴，如下图：

IE6/IE7下依旧并排显示 张鑫旭-鑫空间-鑫生活

OK，现在应该很好理解最上面为何在现代浏览器下图片文字重叠而IE6/IE7下并排显示了。

在“CSS float浮动的深入研究、详解及拓展”系列中多次阐述了浮动元素的“无高度”特性，所以，当图片应用了 `float:left` 属性后，图片所占据的高度丢失，于是，原本换行显示在下面的文字就提上去了，于是就形成了重叠，如下图标示：

浮动与绝对定位重叠图示 张鑫旭-鑫空间-鑫生活

## 四、下节看点

下篇讲relative属性，内容包括：

relative的占位性；

relative最小化影响原则；

relative在一些bug中的修复功能；

等.....

时间仓促，资质有限，文章难免有表述不准确或是理解有误的地方，欢迎指正，不甚感谢。

原创文章，转载请注明来自张鑫旭-鑫空间-鑫生活 [http://www.zhangxinxu.com]

本文地址：<http://www.zhangxinxu.com/wordpress/?p=1528>

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。

标签：[absolute](#), [float](#), [IE6](#), [IE7](#), [IE8](#), [margin](#), [relative](#), [table-cell](#), [visibility](#), [兼容性](#), [显示](#), [等高布局](#)

赞助商推荐 [我也要赞助](#)：





1.

秋山说道：

2016年08月22日 11:28

关于“二、absolute与等高布局”这里面给父容器设置了overflow:hidden,这个在实际使用时不妥，如果我们这个容器里面有一个日历弹层或者其他弹层，超出了父容器的宽高，此时，超出的部分会被隐藏

回复
2.

律讯小前端说道：

2016年03月21日 17:04

你好大神position:absolute+ display:none 和position:absolute + visibility:hidden在性能上还有没有差异？我实验了一下这样都不会产生回流在性能上是不是可以理解为他们都是一样的？

回复
3.

july\_L说道：

2014年11月19日 14:27

第一张图的IE6/7下margin与absolute元素重叠 图片跑到距左侧的位置不知道是什么原因 设置了定位的left：0px;时就乖乖的留在了原来的位子 实现了和设置父标签为span一样的效果

回复
4.

Joyce说道：

2013年10月24日 11:06

那个“绝对定位与登高布局demo”是有问题的，如果你给两个box都加上颜色，你会发现增加其中一个box的高度，另一个并没有增加。我之前看过一篇等高布局博文，采用的是背景色与内容分离纯css实现，并且兼容IE6~7。具体实现是使用float，嵌套div。有一个缺点就是列数n就需要嵌套n层

回复
5.

小菜说道：

2012年07月23日 12:13



在table中，给table或者td设置：position:relative，td内的元素设置：position:absolute失效，求详解

[回复](#)

张 鑫旭说道：

2012年07月23日 16:08



@小菜 td标签不少CSS属性是不支持的

[回复](#)

6. 薇薇说道：

2012年03月27日 14:39



我自己模仿" margin定位元素绝对定位元素重叠的误会"这篇文章写了demo,发现在虚拟机下面的ie6浏览器里不兼容，把代码给朋友试了下，她用ietester测试的没问题。希望楼主有时间可以再看下。

[回复](#)

7. 弄月说道：

2011年05月24日 17:26



请问为什么用absolute设置隐藏，当显示隐藏元素时，无需判断隐藏前的position值？如果元素在隐藏前是relative或本身就是absolute的元素，隐藏后显示却变成static了，会有问题吗？

[回复](#)

张 鑫旭说道：

2011年05月24日 18:42



@弄月 是有这个问题。

[回复](#)

水中月明说道：

2013年08月22日 03:06



可以定义一个类专门用来进行隐藏，放在css声明的最下方

```
.hidden{position: absolute;top:-9999em;}
```

然后通过控制给元素添加这个类或者移除这个类来实现它的隐藏显示  
这样的方式不是挺好的吗？

[回复](#)

8. 海玉说道：

2011年04月28日 14:57



inline-block化的元素有三大特性：包裹性；高宽可定义；图文混排。然而，div, p这类标签应用了position:absolute后，在非IE6/7浏览器下，只有包裹性和高宽可定义这两个特性，但并不支持图片混排，也就是与图片文字在一起的时候会换行。这个观点很新奇，值得细细推敲一番。

[回复](#)

9.

自由人说道：

2011年04月13日 11:01

学习了啊。。。定位这块着实让我头疼哎。。。  
[回复](#)


10.

淘宝购物说道：

2011年03月30日 21:35

膜拜中，请打扰！  
[回复](#)


11.

娃哈哈说道：

2011年03月29日 19:06

能把张含韵小姐的左半边放出来咩~~很期待的说~~  
[回复](#)


12.

直觉说道：

2011年03月25日 16:45

用absolute实现等高布局真的很巧妙哇！  
图文混排也是经常要用到的，这篇文章让人受益匪浅  
[回复](#)


13.

feng33说道：

2011年03月24日 14:21

1px有利于屏幕解读器的正确读取  
[回复](#)


14.

feng33说道：

2011年03月24日 13:49

之前看过确保边框的显示总是用了这样的隐藏模式.hidden { position: absolute !important; clip: rect(1px 1px 1px 1px); /\* IE6, IE7 \*/ clip: rect(1px, 1px, 1px, 1px); padding: 0 !important; border: 0 !important; height: 1px !important; width: 1px !important; overflow: hidden; }  
[回复](#)


15.

wmtimes说道：

2011年03月23日 13:13

相对和绝对定位这个属性用的还比较少。  
[回复](#)


16.

素里太守说道：

2011年03月23日 09:21



“使用display:none与display:block/inline来实现DOM元素的显隐控制，那你就out了”这句话孟浪了，原因听俺慢慢道来。

浏览器的Render Engine(Trident , Gecko , Webkit)中有两个最重要的数据结构：DOM TREE与RENDER TREE。DOM TREE是JAVA SCRIPT程序员能够直接操作的数据，比如遍历，修改，删除，增加DOM TREE的结点，但RENDER TREE则不是JAVA SCRIPT程序员能直接涉足的。RENDER TREE必须通过修改DOM TREE的结点相关属性间接实现，最简单的例子非“display:none与display:block/inline”莫属。一个DOM TREE的结点不一定有对应的RENDER TREE结点（比如script），但display属性为none的节点一定没有RENDER TREE结点。将display属性动态设置为none的节点将引发RENDER TREE的节点删除操作，反之亦然。

当某个应用必须权衡考虑内存耗用，计算时间，带宽消耗的时候，“使用display:none与display:block/inline”就变得很有必要，移动网站设计就是一例。

[回复](#)

17. **gouf说道：**

2011年03月22日 14:52



absolute的bug解析得太好了

[回复](#)

18. **夏雨说道：**

2011年03月21日 19:04



绝对定位等高布局是亮点！

[回复](#)

19. **濯焰说道：**

2011年03月20日 23:18



用absolute实现隐藏，很强大！

[回复](#)





