要想更好的理解 css， 尤其是 IE 下对 css 的渲染，haslayout 是一个非常有必要彻底弄清楚的概念。大多 IE  
下的显示错误，就是源于 haslayout。

**什么是 haslayout ？**

haslayout 是Windows Internet Explorer渲染引擎的一个内部组成部分。在Internet Explorer中，一个元素要么自己对自身的内容进行计算大小和组织，要么依赖于父元素来计算尺寸和组织内容。为了调节这两个不同的概念，渲染引擎采用了 hasLayout 的属性，属性值可以为true或false。当一个元素的 hasLayout 属性值为true时，我们说这个元素有一个布局（layout）

当一个元素有一个布局时，它负责对自己和可能的子孙元素进行尺寸计算和定位。简单来说，这意味着这个元素需要花更多的代价来维护自身和里面的内容，而不是依赖于祖先元素来完成这些工作。因此，一些元素默认会有一个布局。当我们说一个元素“拥有layout”或“得到layout”，或者说一个元素“has layout” 的时候，我们的意思是指它的微软专有属性 hasLayout 被设为了 true 。一个“layout元素”可以是一个默认就拥有 layout 的元素或者是一个通过设置某些 CSS 属性得到 layout 的元素。通过 IE Developer Toolbar 可以查看 IE 下 HTML 元素是否拥有haslayout，在 IE Developer Toolbar 下，拥有 haslayout 的元素，通常显示为“haslayout = -1”。

特别注意的是，hasLayout 在 IE 8 及之后的 IE 版本中已经被抛弃，所以在实际开发中只需针对 IE 8 以下的浏览器为某些元素触发 hasLayout 。

一个元素触发 hasLayout 会影响一个元素的尺寸和定位，这样会消耗更多的系统资源，因此 IE 设计者默认只为一部分的元素触发 hasLayout （即默认有部分元素会触发 hasLayout ，这与 BFC 基本完全由开发者通过特定 CSS 触发并不一样），这部分元素如下：

\* body and html  
\* table, tr, th, td  
\* img  
\* hr  
\* input, button, file, select, textarea, fieldset  
\* marquee  
\* frameset, frame, iframe  
\* objects, applets, embed

**如何激发 haslayout？**

大部分的 IE 显示错误，都可以通过激发元素的 haslayout 属性来修正。可以通过设置 css 尺寸属性(width/height)等来激发元素的 haslayout，使其“拥有布局”。如下所示，通过设置以下 css 属性即可。

\* display: inline-block  
\* height: (任何值除了auto)  
\* float: (left 或 right)  
\* position: absolute  
\* width: (任何值除了auto)  
\* writing-mode: tb-rl  
\* zoom: (除 normal 外任意值)

Internet Explorer 7 还有一些额外的属性(不完全列表):

\* min-height: (任意值)  
\* max-height: (除 none 外任意值)  
\* min-width: (任意值)  
\* max-width: (除 none 外任意值)  
\* overflow: (除 visible 外任意值)  
\* overflow-x: (除 visible 外任意值)  
\* overflow-y: (除 visible 外任意值)  
\* position: fixed

其中 overflow-x 和 overflow-y 是 css3 盒模型中的属性，目前还未被浏览器广泛支持。

对于内联元素(默认即为内联的元素，如 span，或 display:inline; 的元素)，

* width 和 height 只在 IE5.x 下和 IE6 或更新版本的 quirks 模式下触发 hasLayout 。而对于 IE6，如果浏览器运行于标准兼容模式下，内联元素会忽略 width 或 height 属性，所以设置 width 或 height 不能在此种情况下令该元素具有 layout。
* zoom 总是可以触发 hasLayout，但是在 IE5.0 中不支持。建议使用 zoom: 1 来触发元素的 hasLayout 。

具有“layout” 的元素如果同时 display: inline ，那么它的行为就和标准中所说的 inline-block 很类似了：在段落中和普通文字一样在水平方向和连续排列，受 vertical-align 影响，并且大小可以根据内容自适应调整。这也可以解释为什么单单在 IE/Win 中内联元素可以包含块级元素而少出问题，因为在别的浏览器中 display: inline 就是内联，不像 IE/Win 一旦内联元素拥有 layout 还会变成 inline-block。

**haslayout 问题的调试与解决**

当网页在 IE 中有异常表现时，可以尝试激发 haslayout 来看看是不是问题所在。常用的方法是给某元素 css 设定 zoom:1 。使用 zoom:1 是因为大多数情况下，它能在不影响现有环境的条件下激发元素的 haslayout。而一旦问题消失，那基本上就可以判断是 haslayout 的原因。然后就可以通过设定相应的 css 属性来对这个问题进行修正了。建议首先要考虑的是设定元素的 width/height 属性，其次再考虑其他属性。

对 IE6 及更早版本来说，常用的方法被称为霍莉破解(Holly hack)，即设定这个元素的高度为 1% (height:1%;)。需要注意的是，当这个元素的 overflow 属性被设置为 visible 时，这个方法就失效了。或者使用 IE 的条件注释。

对 IE7 来说，最好的方法时设置元素的最小高度为 0 (min-height:0;)。

**haslayout 问题引起的常见 bug**

**IE6 及更低版本的双空白边浮动 bug**

bug 修复: display:inline;

**IE5-6/win 的 3 像素偏移 bug**

bug 修复: \_height:1%;

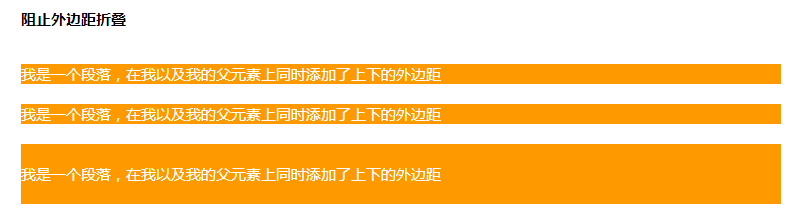
**IE6 的躲躲猫(peek-a-boo) bug**

bug 修复: \_height:1%;

**这里列出触发 hasLayout 元素的一些效果**

**1.阻止外边距折叠**

两个相连的 div 在垂直上的外边距会发生叠加，而触发 hasLayout 的元素之间则不会有这种情况发生，如下图：

[](http://kayosite.com/wp-content/uploads/2012/10/hasLayout-1.png)

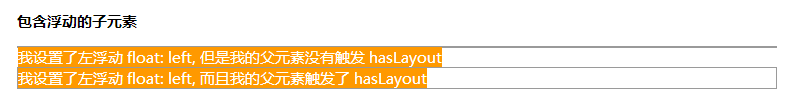
也可以查看 [Demo](http://kayosite.com/demo/hasLayout/index.html#hasLayout-1) 。

上图的例子中，三个 div 各包含一个 p 元素，三个 div 及其包含的 p 元素都有顶部和底部的外边距，但只有第三个 div 的边距没有与它的子元素 p 的外边距折叠。这是因为第三个 div 使用 zoom: 1 触发了 hasLayout ，阻止了它与它的子元素的外边距折叠。

另外，例子中也使用了 overflow: hidden 触发元素的 BFC ，这利用了 BFC 阻止外边距折叠的特性达到元素在 IE 与现代浏览器下的表现统一。

**2.可以包含浮动的子元素，即计算高度时包括其浮动子元素**

效果如图：

[](http://kayosite.com/wp-content/uploads/2012/10/hasLayout-2.png)

也可以看 [Demo](http://kayosite.com/demo/hasLayout/index.html#hasLayout-2) 。

上图的例子中，有两个 div ，它们各包含一个设置了浮动的 p 元素，但第一个 div 实际被浏览器判断为没有高度和宽度，即高度为 0 ，上下边框重叠在一起。而第二个 div 使用 zoom: 1 触发了 hasLayout ，可以包含浮动元素，因此能正确表现出高度，其边框位置也正常了。

本例子中也使用了 overflow: hidden 触发 BFC ，跟上例相似，这利用了 BFC 可以包含浮动子元素的特性达到元素在 IE 与现代浏览器下的表现统一。

**3.背景图像显示问题**

元素背景图不能正确显示是网页重构中最常见的问题之一了，在 IE 7 及以下的 IE 版本中，没有设置高度、宽度的元素往往不能显示出背景图（背景色则显示正常），这实际上与 hasLayout 有关。实际的情况是，没有触发 hasLayout 的元素不能显示背景图，上面有说过，触发 hasLayout 也就是使到元素拥有布局，换句话说，拥有布局的元素才能正确显示背景图。如下图：

[](http://kayosite.com/wp-content/uploads/2012/10/hasLayout-3.png)

也可以看 [Demo](http://kayosite.com/demo/hasLayout/index.html#hasLayout-3) (在 IE 7 或更低版本的 IE 下查看以观察背景图问题)。

上图两个 div 都设置了背景图，但只有使用 zoom: 1 触发了 hasLayout 的第二个 div 才能正确显示背景图。

本例子中没有触发元素的 BFC ，这是因为在现代浏览器中，元素本身并没有背景图显示问题。

可以看出，上面的第一、二个例子中，为了使到元素在 IE (包括低版本 IE 以及较新版本的 IE)和现代浏览器中表现尽量统一同时触发了 hasLayout 和 BFC ，而第三个例子中的问题因为只在低版本 IE 中出现，所以只需为低版本 IE 触发 hasLayout ，这些技巧在实际项目中需要特别注意。

上面也有说道，hasLayout 与很多 IE 下的显示 bugs 都有关，实际上很多莫名奇妙的 bugs 都因 hasLayout 而起，因此只要适当的触发元素的 hasLayout ，很多 IE bugs 往往就能解决。