

你到了第几层？图片式标题、按钮与隐藏文本

这篇文章发布于 2012年03月9日，星期五，16:30，归类于 [CSS相关](#)。阅读 51731 次，今日 2 次 [13 条评论](#)

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

本文地址：<http://www.zhangxinxu.com/wordpress/?p=2265>

一、所谓何物？

所谓“图片式标题/按钮”，是指文字图片化的标题/按钮。比较熟悉的就淘宝网首页了哈～



上面右图（或上图，如果是普屏）的“便民服务”、“卡券&票务”，开小差，忘记圈了～😄

二、你到了第几层？

第一层：面向普通状况下的普通用户

这一层是把图片作为了万能手。举个例子，凡客首页（记得以前凡客首页的选项卡文字都是图片式的）顶部的搜索按钮。我们使用小bug(firebug的爱称)看下源代码：



可以看到，搜索按钮就是个不含任何文本内容的 `input` 标签，no title, no value, no text, 一切的一切都取决于对应的这个background-image.

这是最初级层次的实现，只针对普通状况下的普通用户。所谓普通状况就是网速还凑合，用户没有禁用图片或是图片服务器没有便秘或干脆罢工；所谓普通用户指的就是那些靠眼睛浏览网页的用户。

大家都是聪明人，点到为止。

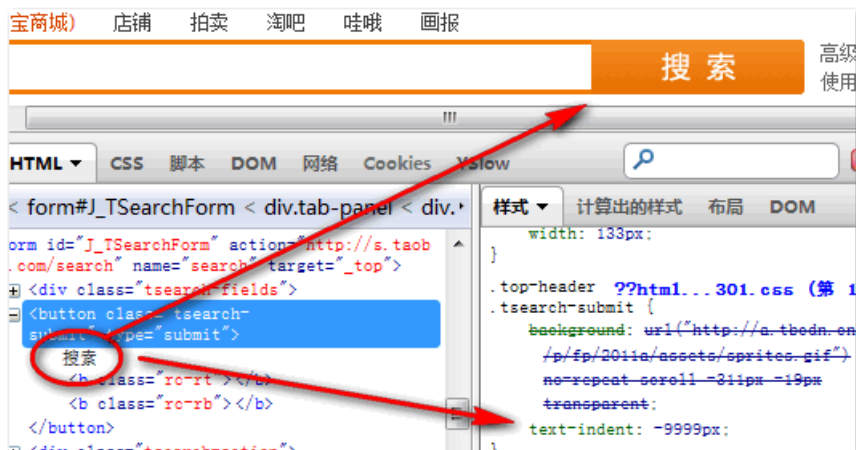
第二层：面向机器、面向开发者，面向普通状况下的普通用户

这一层的用户标签内保留了图片上对应的文字，但是使用的文字隐藏的技术是 `display:none` 或是 `visibility:hidden` 之流。这两个声明隐藏的文字屏幕阅读器是无法抓取的。而且，有可能被搜索引擎酱油化（如果有恶意之嫌）。

说实话，这种层面的做法有时还不如第一层，花费的额外工夫和收效比低。唯一确定的优势就是HTML代码的可读性高了点，便于开发者识别与协作。

第三层：面向机器、面向开发者，面向普通状况下的普通用户以及特殊用户

第三层是标签内保留了图片上对应的文字，通过一些可用性更高的技术隐藏了。例如淘宝首页的搜索：



可以看到。淘宝网首页的搜索添加了“搜索”文字，并通过CSS `text-indent:-9999px`将文字隐藏了。//zxx: 我喜欢使用-999em, 因为可以少一个字符, 哇哇哇! 🍻

虽然 `text-indent` 负值在忘记设置 `overflow:hidden` 声明时候可能会有长长虚框（见下截图，来自淘宝网首页），但是，对于这一层的文字隐藏来说，其为最佳方法。



首先，从SEO的角度讲，文字可以被抓取；开发人员这边，也能迅速确定这段HTML的作用，便于开发协作；对于特殊用户例如使用屏幕阅读器的盲人用户，也能知道这里内容与含义；当然，普通用户正常状态下的访问不受影响。

第四层：面向机器、面向开发者，面向未来的开发者，面向特殊状况、普通状况下的普通用户以及特殊用户

与上一层相比，还是技术实现上的差异。第三层中，文字的隐藏是藏于远方；而这一层，文字的隐藏时藏于文字图片的下方。

隐藏于下方可以应对一些特殊状况，如：

- 开发人员不小心，图片根地址配置还是测试地址
- 因为某些原因，用户禁止了图片的显示
- 因为某些原因，用户禁止了背景图片的显示
- 因为网速的原因图片加载缓慢
- 因为脚本阻塞等原因，图片迟迟不显示
- 图片服务器的DNS解析出了问题
- 图片服务器遭受攻击，反应迟钝，响应缓慢，便秘不断
- 图片服务器做俯卧撑去了~~

- 其他108种影响图片显示的特殊状况.....

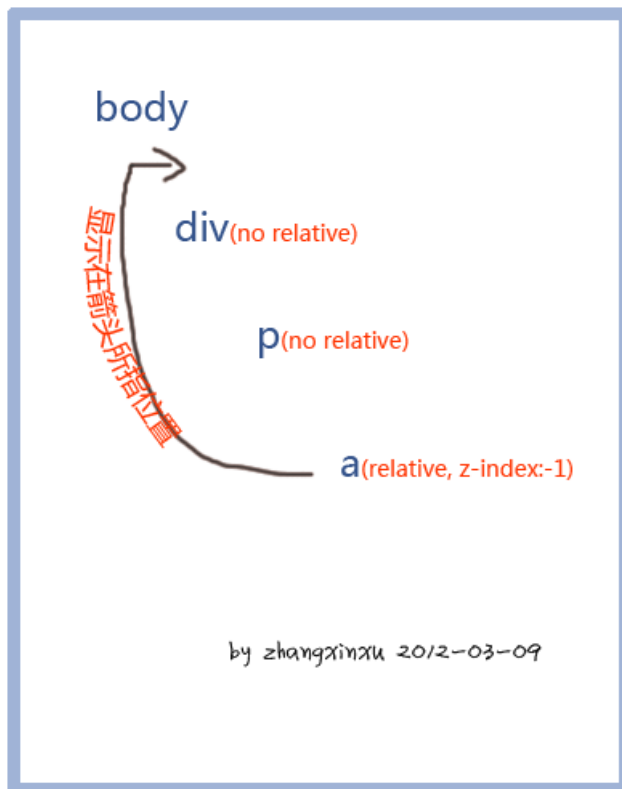
这种情况下，因为文字就在图片的下方，即使图片显示不出来（或延迟显示），文字也会显露出来，丝毫不影响功能使用。这类图片本来就是用来美化与装饰的，怎能左右功能呢？

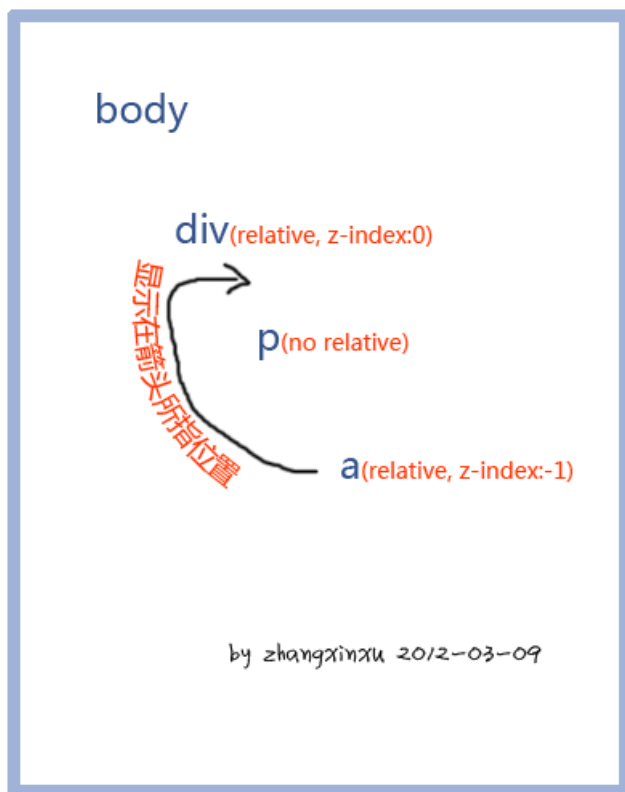
这就是为何说这一层多了个“面向特殊状况下的普通用户”！

至于图片后文本隐藏技术，则是下面要讲的内容。

三、图片后文本隐藏技术之z-index负值隐藏

根据我自己的理解：z-index负值可以让元素在其第一个含有定位属性(relative/absolute等)的父元素之上显示，该父元素其他子元素之下显示。因为 `body` 算是特殊的顶级定位元素，因此，我们基本上可以划分两种情况，见下面手工演示图：





提前需知的~~

下面所有的讨论，示例等都是针对淘宝网首页“我的本地生活”一栏几个彩色标题：



这几个标题是使用 `background-image` 定位以及通过 `text-indent` 负值隐藏文字实现。我们现在要做的就是：保留 `background-image` 定位，文字隐藏使用其他技术。

1. body为定位元素

当要躲猫猫的文字的第一个有定位特性的爸爸就是body的时候，我们直接对文字元素应用：

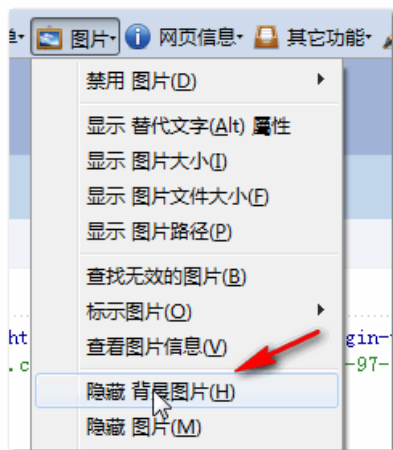
```
position: relative; z-index: -1;
```

就可以实现效果了。您可以狠狠地点击这里：[body为定位元素时文字隐藏技术demo](#)

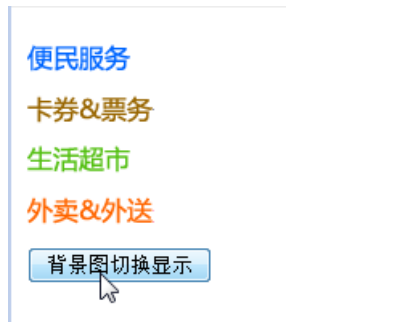
作为普通状态的普通用户的我们进去，看到的可能是个很正常不过的样子，看不到什么猫腻或是玄机。



如果您浏览器有禁用背景图片显示的工具，例如FireFox下的web developer插件：



可以尝试禁用之。如果没有，您也点击demo页面上的“背景图切换显示”按钮，手动隐藏标题的背景图。您会惊奇地看到，后面显示出了对应的文字，如下动图显示：



demo页面有源代码，这里就重复不展示了。

利用body定位的局限性

由于文字藏匿于body之上（而非就在标题图片之下），实现该效果是需要特定条件的。即：含标题背景图片元素到body元素间的父级元素不能有背景色或其他东西遮挡。这个很好理解：依次有a, b, c三栋楼，你住在a楼（background-image为窗帘），想看c楼前的一棵树（隐藏文本），需要的条件是拉开窗帘（无background-image），b楼不挡住视线（无背景色背景图遮挡）。

这里，不得不赞下淘宝的首页，页面的背景色，模块的背景色等都交给了用户（例如我们修改页面背景色为 #beceeb ）：



加上没有先前相对定位属性泛滥的情况，这使得基于body实现z-index负值文本隐藏变得无阻碍。

2. 普通元素为定位元素

然而，考虑到现实，标题图片或按钮图片之上有含背景色的父标签是很正常的。因此，基于body定位很多时候是行不通的。这个时候，我们可以多写点代码，在普通元素上进行z-index文本隐藏。

这里的普通元素指“从文字直系父标签到第一个含有背景的祖先元素间的所有元素（包括这个祖先元素）”。普通元素上添加诸如下面代码即可：

```
position:relative; z-index:0;
```

您可以狠狠地点击这里：[普通元素下z-index文本隐藏demo](#)

//xxx:可能你跟第一个demo看不出什么差异。实际上，第一个demo标题图片所在的白色背景为 `body` 标签设置的背景色（相关设置CSS未展示）；而第二个demo标题图片所在的白色背景为右侧 `id` 为 `effect` 所在的区域的背景色。

对比第一个demo，这里多了点：

```
#effect {  
    position: relative;  
    z-index: 0;  
}
```

效果虽然实现，但是参照“[relative的最小化印象原则](#)”，上面code还可以改进，即CSS最好添加在标题图片元素的直系父标签上。

```
.demo {  
    position: relative;  
    z-index: 0;  
}
```

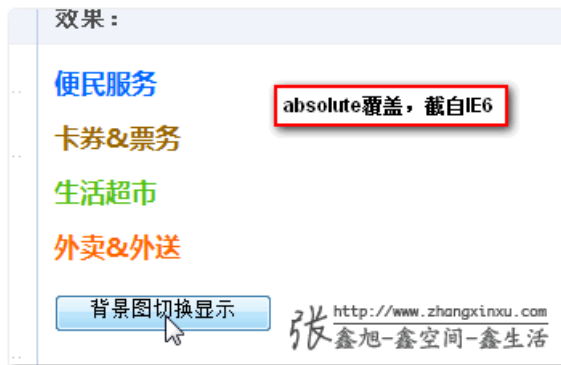
您可以狠狠地点击这里：[进一步的z-index文本隐藏demo](#)

兼容IE6+以及其他阿猫阿狗浏览器。

四、图片后文本隐藏技术之absolute覆盖

这个技术估计知道的人多，懒得多讲。很简单：标题图片或按钮应用 `position:absolute`，覆盖在文字之上。

您可以狠狠地点击这里：[absolute覆盖实现文字隐藏demo](#)



类似下面的HTML结构：

```
<h4 class="title title_1"><a href="#"></a>便民服务</h4>  
<h4 class="title title_2"><a href="#"></a>卡券&票务</h4>  
<h4 class="title title_3"><a href="#"></a>生活超市</h4>  
<h4 class="title title_4"><a href="#"></a>外卖&外送</h4>
```

其中空 `a` 标签就是背景图片所在标签。自然，我们也可以使用 `img` 标签，页面前定位。

五、好吧，结语

本文啰哩吧嗦了这多，其实本意是介绍 `z-index` 负值以及在文本隐藏中的实际应用。然后，扯东扯西

，就这样了。

不是什么新的内容，我相信肯定有不少同行有类似的研究或介绍，自然，也一定有比我这里更好的方法，欢迎分享！

另外，资历有限，文中可能有一些表述不准确之处（例如，body元素是否默认含有定位属性），欢迎指正！



感谢您能够坚持阅读到这里！

最后再插上一句，如果您无法很好地驾驭 `z-index` 负值（一是自身理解上不够深刻，二是平时糟糕的书写习惯使得页面定位元素、层级混乱），我建议您还是老老实实使用absolute覆盖，简单易懂。还有，从实际情况来讲，我们有没有必要为了一点点可用性上的提高如此折腾呢？说句不好听的话，你的站点如果就那么几个闲得慌的人进来看看的话，你做好第一层就够了；如果你非要对得起自己辛苦的那点工资，`text-indent` 这种程度足矣。所谓程序员自我实现的满足对于用户来讲是没有一点意义的。“多大的平台做多大的事”是有其自身道理的~~~~

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

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



« [ol元素及相关属性:type, start, value, 和reversed](#)

[WAI-ARIA无障碍网页应用属性完全展示](#) »

猜你喜欢

- 页面可用性之outline轮廓外框的一些研究
- CSS hover效果的逆向思维实现
- 找到适合自己的前端发展方向
- 说说HTML5中label标签的可访问性问题
- WAI-ARIA无障碍网页应用属性完全展示
- 翻译-盲人如何使用互联网的8个误区
- 基于VoiceOver的移动web站无障碍访问实战
- HTML5扩展之微数据与丰富网页摘要
- 翻译 - 高性能网站需避免的7个错误
- 小tip: 使用meta实现页面的定时刷新或跳转
- span与a元素的键盘聚焦性以及键盘点击性研究

分享到:

0

标签:

absolute, relative, SEO, zIndex负值, 可访问性, 屏幕阅读器, 文字偏移, 背景图片

发表评论（目前13条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. 做一点自己的事说道:

2013年03月17日 15:52

请问, 第三种方法。在父标签设置background-image的情况下

```
#effect {  
position: relative;  
z-index: 0;  
}
```

为什么就可以看到a标签的文本? 文本的z-index为负值, 应该z-index=0在下面, 文本被隐藏了才对啊?

[回复](#)



2. Hooboy说道:

2012年12月7日 11:34

我不推荐用第四种定位方法。

[回复](#)



3. 6^^|^9说道:

2012年11月13日 17:25

第三种要触发普通元素的haslayout属性 (.demo {position: relative;z-index: 0;}) , 不然的话在ie下第一个图标会受到干扰。这是为什么呢?

[回复](#)



4. 小妖说道:

2012年05月20日 20:21

好像没怎么看 懂的说~~

[回复](#)



5. chinaynlmq说道:

2012年04月9日 17:39

旭 哥真厉害一直在关注! 从第一次接触css到现在(本人刚入行)。发现你总走在前面。每一次看到你写的文章到可以学习到很多值得深思的地方。关于: position这个属性众说风云, 目前为止我也只会用relative和absolution

[回复](#)



6. 火晴云说道:

2012年03月28日 17:44

请问我是标题中 给h4加上font-size:0;overflow:hidden;的方法, 也属于这篇文章的第三层吧?

[回复](#)



张鑫旭说道:

2012年03月28日 20:10

@火晴云 `overflow:hidden`; 没有定高的话就是打酱油的; `font-size:0`; Chrome浏览器下需要做些额外功夫才行;

[回复](#)



7. Edison说道:

2012年03月19日 23:09

请问图片后隐藏文本是雅黑字体, 那图片上的字体是啥啊, 挺好看的方方正正的。

[回复](#)



8. alex说道:

2012年03月14日 17:14

貌似 我在用你认为比较挫的第二层

[链接](#)

学习学习~

[回复](#)



9. 行者说道:

2012年03月12日 17:51

学习一下, 利用相对定位实现, 还没有试过。又学到一招。

[回复](#)



10. 猫猫儿说道:

2012年03月9日 23:15

我一般都这样写`text-indent:-999px`。。足够了。。我也尽量让设计师不要搞奇奇怪怪的字体上去。。

[回复](#)



11. Mask说道:

2012年03月9日 20:10

九阳神功第九层啊, 呵呵, 很好滴文章啊, 提醒一下你的demo使用的图片淘宝好像有变动, 和你截图的gif不一样了~ 不过不影响理解

[回复](#)



12. 滴水成冰说道:

2012年03月9日 17:30

学习了 呵呵

[回复](#)



最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果

- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁹³⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹²⁰⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹⁵⁾
- » [HTML5终极备忘大全（图片版+文字版）](#) ⁽⁹³⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸⁶⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸²⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁸⁰⁾
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) ⁽⁷⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁶⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷⁶⁾

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) ⁽⁷⁶⁾
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) ⁽⁶⁴⁾
- » [看，for..in和for..of在那里吵架！](#) ⁽⁶⁰⁾
- » [是时候好好安利下LuLu UI框架了！](#) ⁽⁴⁷⁾
- » [原来浏览器原生支持JS Base64编码解码](#) ⁽³⁵⁾
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) ⁽³³⁾
- » [炫酷H5中序列图片视频化播放的高性能实现](#) ⁽³¹⁾
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) ⁽³⁰⁾
- » [windows系统下批量删除OS X系统.DS_Store文件](#) ⁽²⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽²⁶⁾

猜你喜欢

- [页面可用性之outline轮廓外框的一些研究](#)
- [CSS hover效果的逆向思维实现](#)
- [找到适合自己的前端发展方向](#)
- [说说HTML5中label标签的可访问性问题](#)
- [WAI-ARIA无障碍网页应用属性完全展示](#)
- [翻译-盲人如何使用互联网的8个误区](#)
- [基于VoiceOver的移动web站无障碍访问实战](#)
- [HTML5扩展之微数据与丰富网页摘要](#)
- [翻译 - 高性能网站需避免的7个错误](#)
- [小tip: 使用meta实现页面的定时刷新或跳转](#)
- [span与a元素的键盘聚焦性以及键盘点击性研究](#)