# [CSS Sticky Footer: 完美的CSS绝对底部](http://blog.csdn.net/jacketinsysu/article/details/44513835)

标签： [css](http://www.csdn.net/tag/css)[footer](http://www.csdn.net/tag/footer)

2015-03-21 10:20 1470人阅读 [评论](http://blog.csdn.net/jacketinsysu/article/details/44513835#comments)(0) [收藏](http://blog.csdn.net/jacketinsysu/article/details/javascript:void(0);) [举报](http://blog.csdn.net/jacketinsysu/article/details/44513835#report)

 分类：



前端（1）



原文：http://paranimage.com/css-sticky-footer/

CSS的简单在于它易学，CSS的困难在于寻找更好的解决方案。在CSS的世界里，似乎没有完美这种说法。所以，现在介绍的CSS绝对底部，只是目前个人见过的方案中比较完美的吧。

先说我们为什么会使用到这个CSS底部布局解决方案:

当做一个页面时，如果页面内容很少，不足于填充一屏的窗口区域，按普通的布局，就会出现下面图片中的样子(也就是底部内容并没有位于窗口的底部，而留下了大量空白。



对于追未完美的设计师来说，这是不美观的。网上有一些解决方案，但会出现当改变窗口高度时，底部和正文重叠的BUG。尽管没有多少人会有事没事儿的去改变窗口高度，但设计嘛，追求的就是尽善尽美。



下面是我找到的一个比较完美的方法，来自国外的设计达人，纯CSS，可以实现: 当正文内容很少时，底部位于窗口最下面。当改变窗口高度时，不会出现重叠问题。



甚至，创造该CSS的人还专门成立一个网站介绍这个CSS底部布局方案。不知道他有没有去申请专利:)

### 代码写法

HTML代码:

<div id="wrap">

<div id="main" class="clearfix">

<div id="content">

</div>

<div id="side">

</div>

</div>

</div>

<div id="footer">

</div>

说明: 使用这个布局的前提，就是footer要在总的div容器之外，footer使用一个层，其它所有内容使用一个总的层。如果确实需要到添加其它同级层，那这个同级层就必须使用position:absolute进行绝对定位。

CSS代码:

下面是主要的CSS代码，让你的底部可以位于窗口的最下面:

html, body, #wrap {height: 100%;}

body > #wrap {height: auto; min-height: 100%;}

#main {padding-bottom: 150px;} /\* 必须使用和footer相同的高度 \*/

#footer {position: relative;

margin-top: -150px; /\* footer高度的负值 \*/

height: 150px;

clear:both;}

说明: 需要注意的就是#main的padding值、footer的高度和负margin值，需要保持一致。

就是这么简单，不过还没完。如果你的主体是使用悬浮布局，还得解决一些浏览器的兼容问题，这里使用的重点是为了Goolge Chrome。

对#main部份进行著名的Clearfix Hack:

.clearfix:after {content: ".";

display: block;

height: 0;

clear: both;

visibility: hidden;}

.clearfix {display: inline-block;}

/\* Hides from IE-mac \\*/

\* html .clearfix { height: 1%;}

.clearfix {display: block;}

/\* End hide from IE-mac \*/

注: 该方案测试于两栏悬浮布局，兼容各大主流浏览器，包括[Google Chrome](http://paranimage.com/tag/google-chrome/)。

P.S:

* 网络上之前比较知名的footer布局有[Ryan Faits](http://ryanfait.com/resources/footer-stick-to-bottom-of-page/)创造的，不过它的方法在HTML代码中会有一个空的div层。严格来说，是不符合语义网代码标准的。
* 另外，还有一篇[Exploring Footers article from A List Apart](http://www.alistapart.com/articles/footers/)，但使用了一些JavaScript代码。
* 这样一比较，上面看似简单的纯CSS，就显得伟大许多了。

OK, 没有了。如果没看懂，具体的使用方法和说明可以[到原站查看](http://www.cssstickyfooter.com/using-sticky-footer-code.html)。

# 使用 position:sticky 实现粘性布局

2017/02/16 · [CSS](http://web.jobbole.com/category/css/) · [position](http://web.jobbole.com/tag/position/), [布局](http://web.jobbole.com/tag/%e5%b8%83%e5%b1%80/)

分享到：<http://www.jiathis.com/share?uid=1745061>**3**

本文作者： [伯乐在线](http://web.jobbole.com/) - [chokcoco](http://www.jobbole.com/members/chokcoco) 。未经作者许可，禁止转载！  
欢迎加入伯乐在线 [专栏作者](http://blog.jobbole.com/99322)。

如果问，CSS 中 position 属性的取值有几个？  
大部分人的回答是，大概是下面这几个吧？



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | {      position: static;      position: relative;      position: absolute;      position: fixed;  } |

额，其实，我们还可以有这 3 个取值：



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | {  */\* 全局值 \*/*      position: inherit;      position: initial;      position: unset;  } |

没了吗？偶然发现其实还有一个处于实验性的取值，position:sticky（[戳我查看MDN解释](https://developer.mozilla.org/zh-CN/docs/Web/CSS/position)）：



|  |  |
| --- | --- |
| 1  2  3 | {      position: sticky;  } |

卧槽，什么来的？



前端发展太快，新东西目接不暇，但是对于有趣的东西，还是忍不住一探究竟。（只怪当初…）

## 初窥 position:sticky

sticky 英文字面意思是粘，粘贴，所以姑且称之为粘性定位。下面就来了解下这个处于实验性的取值的具体功能及实用场景。

这是一个结合了 position:relative 和 position:fixed 两种定位功能于一体的特殊定位，适用于一些特殊场景。

什么是结合两种定位功能于一体呢？

元素先按照普通文档流定位，然后相对于该元素在流中的 flow root（BFC）和 containing block（最近的块级祖先元素）定位。

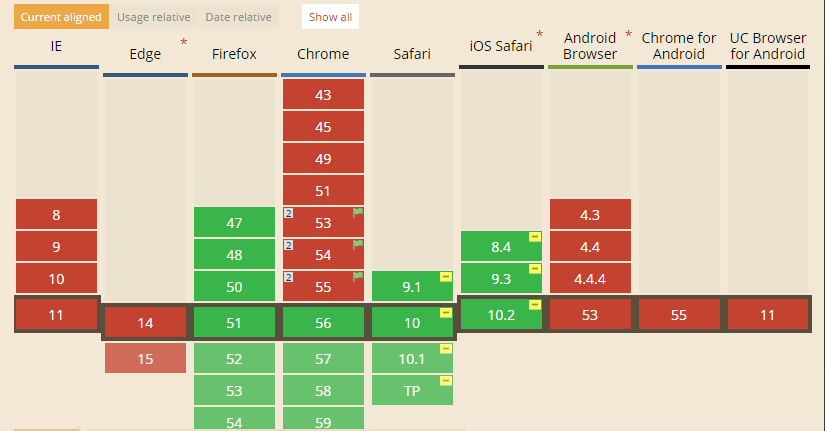
而后，元素定位表现为在跨越特定阈值前为相对定位，之后为固定定位。

这个特定阈值指的是 top, right, bottom 或 left 之一，换言之，指定 top, right, bottom 或 left 四个阈值其中之一，才可使粘性定位生效。否则其行为与相对定位相同。

## 不乐观的兼容性

在讲述具体示例之前，还是很有必要了解一下 position:sticky 的兼容性，嗯，不乐观的兼容性。

看看 [CANIUSE](http://caniuse.com/#feat=css-sticky) 下的截图：



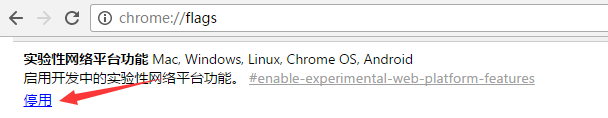
SHIT，这么好的属性支持性居然这么惨淡。



IOS 家族（SAFARI && IOS SAFARI）和 Firefox 很早开始就支持 position:sticky 了。而 Chrome53~55 则需要启用实验性网络平台功能才行。其中 webkit 内核的要添加上私有前缀 -webkit-。

### Chrome53~55 开启实验性网络平台功能

地址栏输入 chrome://flags/ ，找到 enable-experimental-web-platform-features ，选择启用：



所以下面的 CodePen 示例，需要上述几个浏览器下观看。

## position:sticky 示例

嗯，上面的文字描述估计还是很难理解，看看下面这张 GIF 图，想想要实现的话，使用 JS + CSS 的方式该如何做：



按照常规做法，大概是监听页面 scroll 事件，判断每一区块距离视口顶部距离，超过了则设定该区块 position:fixed，反之去掉。

而使用 position:sticky ，则可以非常方便的实现（请在 SAFARI 或者 CHROME53+ 下观看）：

See the Pen [positionSticky](http://codepen.io/Chokcoco/pen/XpGjJg/) by Chokcoco ([@Chokcoco](http://codepen.io/Chokcoco)) on [CodePen](http://codepen.io/).

嗯，看看上面的 CSS 代码，只需要给每个内容区块加上



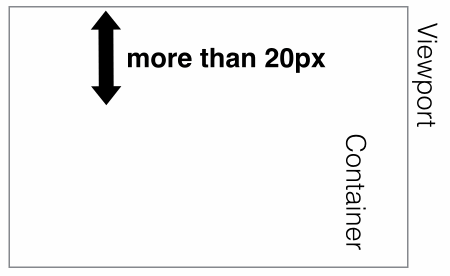
|  |  |
| --- | --- |
| 1  2  3  4  5 | {      position: -webkit-sticky;      position: sticky;      top: 0;  } |

就可以轻松实现了。

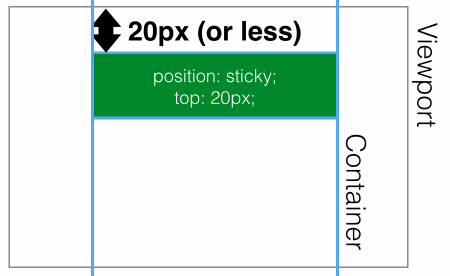
简单描述下生效过程，因为设定的阈值是 top:0 ，这个值表示当元素距离页面视口（Viewport，也就是fixed定位的参照）顶部距离大于 0px 时，元素以 relative 定位表现，而当元素距离页面视口小于 0px 时，元素表现为 fixed 定位，也就会固定在顶部。

不理解可以再看看下面这两张示意图（top:20px 的情况，取自开源项目[fixed-sticky](https://github.com/filamentgroup/fixed-sticky)）：

#### 距离页面顶部大于20px，表现为 position:relative;

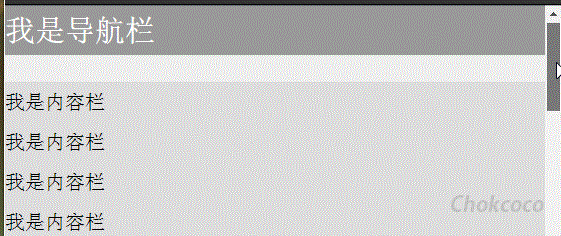


#### 距离页面顶部小于20px，表现为 position:fixed;



## 运用 position:sticky 实现头部导航栏固定

运用 position:sticky 实现导航栏固定，也是最常见的用法：



（请在 SAFARI 或者 CHROME53+ 下观看）：

See the Pen [stickyNav](http://codepen.io/Chokcoco/pen/OWqpjJ/) by Chokcoco ([@Chokcoco](http://codepen.io/Chokcoco)) on [CodePen](http://codepen.io/).

同理，也可以实现侧边导航栏的超出固定。

## 生效规则

position:sticky 的生效是有一定的限制的，总结如下：

1. 须指定 top, right, bottom 或 left 四个阈值其中之一，才可使粘性定位生效。否则其行为与相对定位相同。
   * 并且 top 和 bottom 同时设置时，top 生效的优先级高，left 和 right 同时设置时，left 的优先级高。
2. 设定为 position:sticky 元素的任意父节点的 overflow 属性必须是 visible，否则 position:sticky 不会生效。这里需要解释一下：
   * 如果 position:sticky 元素的任意父节点定位设置为 overflow:hidden，则父容器无法进行滚动，所以 position:sticky 元素也不会有滚动然后固定的情况。
   * 如果 position:sticky 元素的任意父节点定位设置为 position:relative | absolute | fixed，则元素相对父元素进行定位，而不会相对 viewprot 定位。
3. 达到设定的阀值。这个还算好理解，也就是设定了 position:sticky 的元素表现为 relative 还是 fixed 是根据元素是否达到设定了的阈值决定的。

## 开始使用？

上面从兼容性也看到了，情况不容乐观，但是用于某些布局还是能省很多力的，如果真的希望用上这个属性，可以采用一些开源库来实现兼容。

推荐 [fixed-sticky](https://github.com/filamentgroup/fixed-sticky) 。

系列 CSS 文章汇总在我的 [Github](https://github.com/chokcoco/iCSS) 。

到此本文结束，如果还有什么疑问或者建议，可以多多交流，原创文章，文笔有限，才疏学浅，文中若有不正之处，万望告知。