

CSS Position(定位)

position 属性指定了元素的定位类型。

position 属性的五个值：

- static
- relative
- fixed
- absolute
- sticky

元素可以使用的顶部，底部，左侧和右侧属性定位。然而，这些属性无法工作，除非是先设定position属性。他们也有不同的工作方式，这取决于定位方法。

static 定位

HTML 元素的默认值，即没有定位，遵循正常的文档流对象。

静态定位的元素不会受到 top, bottom, left, right影响。

实例

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

[尝试一下 »](#)

fixed 定位

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移动：

实例

```
p.pos_fixed  
{  
  position:fixed;  
  top:30px;  
  right:5px;  
}
```

[尝试一下 »](#)

注意：Fixed 定位在 IE7 和 IE8 下需要描述 !DOCTYPE 才能支持。

Fixed定位使元素的位置与文档流无关，因此不占据空间。

Fixed定位的元素和其他元素重叠。

relative 定位

相对定位元素的定位是相对其正常位置。

实例

```
h2.pos_left
{
  position:relative;
  left:-20px;
}
h2.pos_right
{
  position:relative;
  left:20px;
}
```

[尝试一下 »](#)

移动相对定位元素，但它原本所占的空间不会改变。

实例

```
h2.pos_top
{
  position:relative;
  top:-50px;
}
```

[尝试一下 »](#)

相对定位元素经常被用来作为绝对定位元素的容器块。

absolute 定位

绝对定位的元素的位置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>:

实例

```
h2
{
  position:absolute;
  left:100px;
  top:150px;
}
```

[尝试一下 »](#)

absolute 定位使元素的位置与文档流无关，因此不占据空间。

absolute 定位的元素和其他元素重叠。

sticky 定位

sticky 英文字面意思是粘，粘贴，所以可以把它称之为粘性定位。

position: sticky; 基于用户的滚动位置来定位。

粘性定位的元素是依赖于用户的滚动，在 **position:relative** 与 **position:fixed** 定位之间切换。

它的行为就像 **position:relative**；而当页面滚动超出目标区域时，它的表现就像 **position:fixed**；它会固定在目标位置。

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。

注意：Internet Explorer, Edge 15 及更早 IE 版本不支持 sticky 定位。Safari 需要使用 -webkit- prefix (查看以下实例)。

实例

```
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```

尝试一下 »

重叠的元素

元素的定位与文档流无关，所以它们可以覆盖页面上的其它元素

z-index属性指定了一个元素的堆叠顺序（哪个元素应该放在前面，或后面）

一个元素可以有正数或负数的堆叠顺序：

实例

```
img  
{  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

尝试一下 »

具有更高堆叠顺序的元素总是在较低的堆叠顺序元素的前面。

注意：如果两个定位元素重叠，没有指定 z - index，最后定位在HTML代码中的元素将被显示在最前面。



更多实例

裁剪元素的外形

此示例演示如何设置元素的外形。该元素被剪裁成这种形状，并显示出来。

如何使用滚动条来显示元素内溢出的内容

这个例子演示了overflow属性创建一个滚动条，当一个元素的内容在指定的区域过大时如何设置以适应。

如何设置浏览器自动溢出处理

这个例子演示了如何设置浏览器来自动处理溢出。

更改光标

这个例子演示了如何改变光标。

所有的CSS定位属性

"CSS" 列中的数字表示哪个CSS(CSS1 或者CSS2)版本定义了该属性。

属性	说明	值	CSS
bottom	定义了定位元素下外边距边界与其包含块下边界之间的偏移。	auto <i>length</i> % inherit	2
clip	剪辑一个绝对定位的元素	<i>shape</i> auto inherit	2
cursor	显示光标移动到指定的类型	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text	2

		wait help	
left	定义了定位元素左外边距边界与其包含块左边界之间的偏移。	auto <i>length</i> % inherit	2
overflow	设置当元素的内容溢出其区域时发生的事情。	auto hidden scroll visible inherit	2
overflow-y	指定如何处理顶部/底部边缘的内容溢出元素的内容区域	auto hidden scroll visible no-display no-content	2
overflow-x	指定如何处理右边/左边边缘的内容溢出元素的内容区域	auto hidden scroll visible no-display no-content	2
position	指定元素的定位类型	absolute fixed relative static inherit	2
right	定义了定位元素右外边距边界与其包含块右边界之间的偏移。	auto <i>length</i> % inherit	2
top	定义了一个定位元素的上外边距边界与其包含块上边	auto	2

	界之间的偏移。	<i>length</i> % inherit	
z-index	设置元素的堆叠顺序	<i>number</i> auto inherit	2

[← CSS Display\(显示\) 与 Visibility \(可见性\)](#)
[CSS Float \(浮动\) →](#)


2 篇笔记

[写笔记](#)


relative：定位是相对于自身位置定位（设置偏移量的时候，会相对于自身所在的位置偏移）。设置了 relative 的元素仍然处在文档流中，元素的宽高不变，设置偏移量也不会影响其他元素的位置。最外层容器设置为 relative 定位，在没有设置宽度的情况下，宽度是整个浏览器的宽度。

absolute：定位是相对于离元素最近的设置了绝对或相对定位的父元素决定的，如果没有父元素设置绝对或相对定位，则元素相对于根元素即 html 元素定位。设置了 absolute 的元素脱离了文档流，元素在没有设置宽度的情况下，宽度由元素里面的内容决定。脱离后原来的位置相当于是空的，下面的元素会来占据位置。

浮桥人生 11个月前 [04-12]



postion:

- 1、**absolute(绝对定位)**，其位置相对于最近已定位的父元素，如果元素没有已定位的父元素那么它的位置相对于
- 2、**static(默认的静态定位)**，即没有定位，遵循正常的文档流对象，静态定位的元素不受top、left、right、bottom影响。
- 3、**relative(相对定位)**，其位置相对其正常时的位置。相对定位元素经常被用来作为绝对定位元素的容器块。
- 4、**fixed**，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移动。
- 5、**sticky(粘性定位)**，基于用户滚动位置来定位，在未滚动出目标区域时，它的行为就像 position:relative;它的表现就像 position:fixed;，它会固定在目标位置。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。

咩咩 1周前 [03-06]

