**CSS Position(定位)**

**position的应用场景：侧边栏，头部固定，二级菜单，重叠元素**

position 属性指定了元素的定位类型。

position 属性的五个值：

* [static](https://www.runoob.com/css/css-positioning.html#position-static)
* [relative](https://www.runoob.com/css/css-positioning.html#position-relative)
* [fixed](https://www.runoob.com/css/css-positioning.html#position-fixed)
* [absolute](https://www.runoob.com/css/css-positioning.html#position-absolute)
* [sticky](https://www.runoob.com/css/css-positioning.html#position-sticky)(这个用的很少)

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 »**](https://www.runoob.com/try/try.php?filename=trycss_position_static)

**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 »**](https://www.runoob.com/try/try.php?filename=trycss_position_fixed)

**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。

固定定位的元素可以设置位置偏移量，它是参考document来偏移的。

## 实例

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>      <div class="box1"></div>      <div class="box2"></div>      <div class="box3"></div>  </body>  </html> |
| 效果： |

## 可以设置固定定位的偏移量

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/              left: 100px;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>      <div class="box1"></div>      <div class="box2"></div>      <div class="box3"></div>  </body>  </html> |
|  |

## 定位偏移参数右left，top，right，bottom，其中left和top的优先级比right和bottom的优先级高如果把四个参数都设置为0，元素会固定在左上角

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>      <div class="box1"></div>      <div class="box2"></div>      <div class="box3"></div>  </body>  </html> |
| 效果： |

## 实例实现元素的靠右垂直居中显示

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;              position: fixed;              /\* 设置定位的偏移量 \*/              /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              margin: auto ;/\*上面的简写\*/              right: 0;              top: 0;              bottom: 0;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>      <div class="box1"></div>      <div class="box2"></div>      <div class="box3"></div>  </body>  </html> |
| 效果： |

## 如果此时你把left也设置为0，它就会水平和垂直都居中

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;              position: fixed;              /\* 设置定位的偏移量 \*/              /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              margin: auto ;/\*上面的简写\*/              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>      <div class="box1"></div>      <div class="box2"></div>      <div class="box3"></div>  </body>  </html> |
|  |

**relative 定位**

相对定位元素的定位是相对其正常位置。是参考它原本的位置来偏移的

##### 相对定位不会脱离文档流，这是它和固定定位和绝对定位不一样的地方

**实例**

h2.pos\_left {

position:relative;

left:-20px;

}

h2.pos\_right {

position:relative;

left:20px;

}

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; }

实例1

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              /\* position: fixed; \*/              /\* 相对定位 \*/              position: relative;              /\* 设置定位的偏移量 \*/               /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              /\* margin: auto \*/              left: 100px;              /\* right: 0; \*/              top: 100px;              /\* bottom: 0; \*/          }          .box2{               width:  200px;              height: 200px;              background-color: blue;              }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>          <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div>      </body>  </html> |
|  |

## 实例2：偏移量可以是负的

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              /\* position: fixed; \*/              /\* 相对定位 \*/              position: relative;              /\* 设置定位的偏移量 \*/               /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              /\* margin: auto \*/              left: -50px;              /\* right: 0; \*/              top: -50px;              /\* bottom: 0; \*/          }          .box2{               width:  200px;              height: 200px;              background-color: blue;            }          .box3{              width:  300px;              height: 300px;              background-color: green;          }      </style>  </head>  <body>          <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div>    </body>  </html> |
|  |

##### 注意：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，它会一直往上找，如果都找不到，那么它的位置相对于文档:

绝对定位也会脱离文档流，也是参考整个文档进行定位，也可以使用偏移来控制它的位置，但是它的位置不是固定的，如果有滚动条，他会随着滚动条滚动，这就是他和固定定位的最大区别

**实例**

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

[**尝试一下 »**](https://www.runoob.com/try/try.php?filename=trycss_position_absolute)

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。

#### 实例1：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/               /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              margin: auto ;/\*上面的简写\*/              left: 0;              /\* right: 0; \*/              top: 0;              bottom: 0;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;               /\* 绝对定位 \*/              position: absolute;              margin: auto ;              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }          .out{              position: fixed;              border: 1px solid lime;              width: 700px;          }      </style>  </head>  <body>          <!-- <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div> -->      <div class="out">          <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div>      </div>    </body>  </html> |
|  |

#### 实例2：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>css中的定位属性position</title>      <style>          .box1{              width: 100px;              height: 100px;              background-color: red;              /\* 静态定位 ,是css默认的定位方式\*/              /\* position:static; \*/              /\* 固定定位 \*/              position: fixed;              /\* 设置定位的偏移量 \*/               /\*margin: auto auto;\*//\*如果只写这一句只能实现水平居中，只有结合下面几句才能实现垂直居中和它上面的fixed\*/              margin: auto ;/\*上面的简写\*/              left: 0;              /\* right: 0; \*/              top: 0;              bottom: 0;          }          .box2{              width:  200px;              height: 200px;              background-color: blue;               /\* 绝对定位 \*/              position: absolute;              margin: auto ;              left: 0;              right: 0;              top: 0;              bottom: 0;          }          .box3{              width:  300px;              height: 300px;              background-color: green;          }          .out{              /\* position: fixed; \*/              border: 1px solid lime;              width: 700px;          }      </style>  </head>  <body>          <!-- <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div> -->      <div class="out">          <div class="box1"></div>          <div class="box2"></div>          <div class="box3"></div>      </div>    </body>  </html> |
|  |

##### 注意：绝对定位需要搭配相对定位来使用的，如果儿子使用绝对定位，那么它的父亲需要使用相对定位

**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; }

**定位的层级:z-index**

**语法：{z-index:num;}**

#### num越大，显示层级越高，越显示在上面

#### 必须设置给有定位的元素（static定位除外）

**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 »**](https://www.runoob.com/try/try.php?filename=trycss_zindex)

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](https://www.runoob.com/try/try.php?filename=trycss_clip)

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](https://www.runoob.com/try/try.php?filename=trycss_overflow)

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](https://www.runoob.com/try/try.php?filename=trycss_pos_overflow_auto)

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](https://www.runoob.com/try/try.php?filename=trycss_cursor)

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](https://www.runoob.com/cssref/pr-pos-bottom.html) | 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 *length %* inherit | 2 |
| [clip](https://www.runoob.com/cssref/pr-pos-clip.html) | 剪辑一个绝对定位的元素 | *shape* auto inherit | 2 |
| [cursor](https://www.runoob.com/cssref/pr-class-cursor.html) | 显示光标移动到指定的类型 | *url* auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help | 2 |
| [left](https://www.runoob.com/cssref/pr-pos-left.html) | 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 *length %* inherit | 2 |
| [overflow](https://www.runoob.com/cssref/pr-pos-overflow.html) | 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](https://www.runoob.com/css/cssref/css3-pr-overflow-y.html) | 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](https://www.runoob.com/css/cssref/cssref/css3-pr-overflow-x.html) | 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](https://www.runoob.com/cssref/pr-class-position.html) | 指定元素的定位类型 | absolute fixed relative static inherit | 2 |
| [right](https://www.runoob.com/cssref/pr-pos-right.html) | 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 *length %* inherit | 2 |
| [top](https://www.runoob.com/cssref/pr-pos-top.html) | 定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。 | auto *length %* inherit | 2 |
| [z-index](https://www.runoob.com/cssref/pr-pos-z-index.html) | 设置元素的堆叠顺序 | *number* auto inherit | 2 |

## CSS fixed固定定位transofrm失效及居中小技巧

<https://www.zhangxinxu.com/wordpress/?p=10707> 鑫空间-鑫生活

### 一、经典居中套路

对于绝对定位或者固定定位的元素，如果希望元素水平居中，是有个通用的使用公式的。

如下：

.target {

position: fixed;

left: 50%;

transform: translateX(-50%);

}

如果项目对兼容性要求不高，还可以直接使用translate属性代替：

.target {

position: fixed;

left: 50%;

/\* Chrome 104+, Safari 14.5+ \*/

translate: -50%;

}

但上面的实现有个问题，那就是定位元素的默认可分配的宽度只有50%的屏幕宽度，内容如果超出，就会发生换行，例如：

<span class="target">

春节前上架的掘金小册《技术写作指南》已经2000多人参与学习啦，感谢大家的支持

</span>

效果如下所示（内嵌iframe，实时渲染效果，无内容请访问“鑫空间”[原文](https://www.zhangxinxu.com/wordpress/?p=10707)）：

可以看到，内容显示宽度只有一半，实际期望的效果是内容接近屏幕宽度的时候换行。

使用传统的CSS属性还不太好处理，无论是max-width最大宽度，min-width最小宽度都不合适。

此时，可以试试使用下面的 fit-content 加 margin 方法实现居中对齐。

### 二、fit-content与元素居中

不哔哔，直接上代码，就是下面这样，老少皆宜，童叟无欺。

.target {

position: fixed;

left: 1rem; right:1rem;

width: fit-content;

margin-inline: auto;

}

此时效果如下所示（内嵌iframe，实时渲染，点击文字可连续变换内容）：

至于为何可以居中，你先要了解width:fit-content的含义，《CSS新世界》这本书有说明，虽然固定定位元素默认表现就是fit-content，但是必须显式地声明width属性值，这样，left/right同时设置的时候尺寸才不会拉伸，也给margin:auto的计算留下的空间，居中计算原理《CSS世界》这本书有介绍。

如果手头没有这两本优质CSS书籍也，可以访问“[理解CSS fit-content等width值](https://www.zhangxinxu.com/wordpress/2016/05/css3-width-max-contnet-min-content-fit-content/)”和“[margin:auto实现绝对定位元素的水平垂直居中](https://www.zhangxinxu.com/wordpress/?p=3794)”这两篇历史文章了解相关实现。

总之，绝对定位或固定定位元素的水平居中效果的最佳实现就他了，特别适合用在toast提示这类内容动态的效果中。

### 三、transform与fixed失效的问题

固定定位元素和transform同时使用，元素的固定定位特性是保留的，可要是固定定位元素的祖先有transform设置（或者scale/translate/rotate/filter等属性），则固定定位元素会表现为绝对定位一般。

比方说下面的代码案例：

<div class="outer">

<div class="inner">

<span class="target">技术写作指南，不仅是写作指南</span>

</div>

</div>

.outer {

border: solid;

height: 200px;

overflow: auto;

}

.inner {

height: 600px;

transform: scale(1);

}

.target {

position: fixed;

top: 100px;

}

此时，span.target 元素明明设置的是position:fixed，但依然可以轻松被滚走，就是因为父元素设置了transform:scale(1)。

也可以狠狠地点击这里体验：[transform内的fixed元素变成绝对定位demo](https://www.zhangxinxu.com/study/202302/css-transform-fixed-to-absolute-demo.php)

那有没有什么办法设置transform的同时让子元素依然有固定定位的效果呢？

如果滚动容器是全屏滚动的，则可以使用 position:sticky 黏性定位模拟固定定位的效果。

#### 模拟固定定位

先看演示，您可以狠狠地点击这里：[transform容器内元素依然固定定位demo](https://www.zhangxinxu.com/study/202302/css-transform-still-fixed-demo.php)

这是最终实现的效果，看到没，滚动的时候，目标元素纹丝不动。

相关HTML和CSS代码如下所示：

<div class="wrap">

<span class="target">技术写作指南，不仅是写作指南</span>

</div>

.wrap {

position: absolute;

inset: 0;

width: fit-content;

margin-inline: auto;

transform: scale(1);

pointer-events: none;

}

.target {

position: sticky;

top: 200px;

pointer-events: auto;

}

实现的关键就是给原本需要固定定位的元素外面包裹一层“安全屋”元素，这个元素绝对定位（不占据布局空间），同时高度需要是滚动主体元素的高度（非滚动容器高度），这个时候，粘性定位元素就表现为滚动定位效果了。

更进一步的原理可以参见我之前的这篇文章“[深入理解position sticky粘性定位的计算规则](https://www.zhangxinxu.com/wordpress/2020/03/position-sticky-rules/)”，可能有点难懂，需要多阅读几遍，在《CSS新世界》中也有相关的原理说明。