

专题四: 定位和层叠上下文

定位常见值及原理

position fixed和sticky

层叠上下文和层叠顺序

层叠上下文形成条件

层叠上下文对z-index的影响

大纲目录:

- 定位常见值及原理
- position fixed和sticky
- 层叠上下文和层叠顺序
- 层叠上下文形成条件
- 层叠上下文对z-index的影响

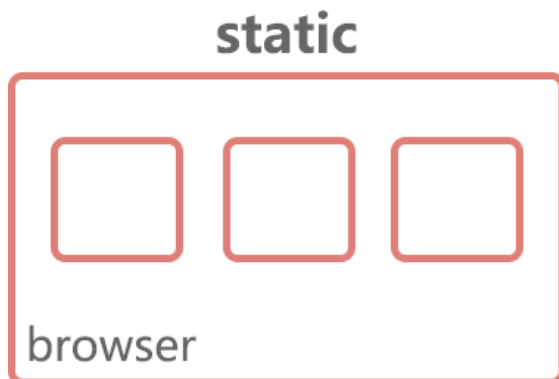
定位常见值及原理

<http://www.ruanyifeng.com/blog/2019/11/css-position.html>

position属性用来指定一个元素在网页上的位置，一共有5种定位方式，即position属性主要有五个值。

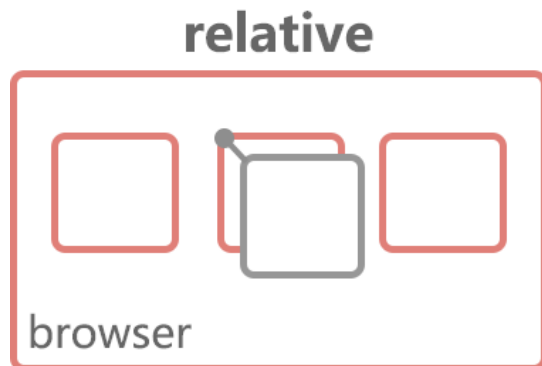
static (默认值):

- static是position属性的默认值。如果省略position属性，浏览器就认为该元素是static定位。
- 这时浏览器会按照源码的顺序，决定每个元素的位置，这称为"正常的页面流"（normal flow）。每个块级元素占据自己的区块（block），元素与元素之间不产生重叠，这个位置就是元素的默认位置。
- 该关键字指定元素使用正常的布局行为，即元素在文档常规流中当前的布局位置。此时 top, right, bottom, left 和 z-index 属性无效;



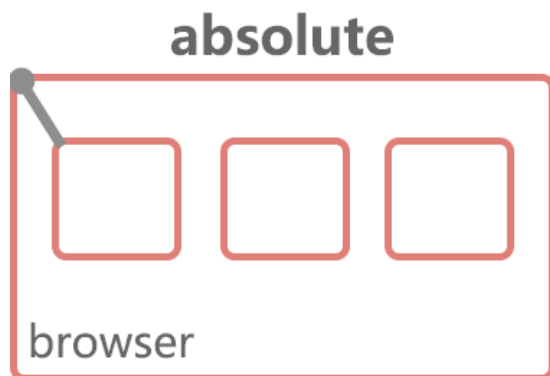
relative(相对定位):

- 表示, 相对于默认位置 (即static时的位置) 进行偏移
- 即定位基点是元素的默认位置
- 它必须搭配top、bottom、left、right这四个属性一起使用, 用来指定偏移的方向和距离;



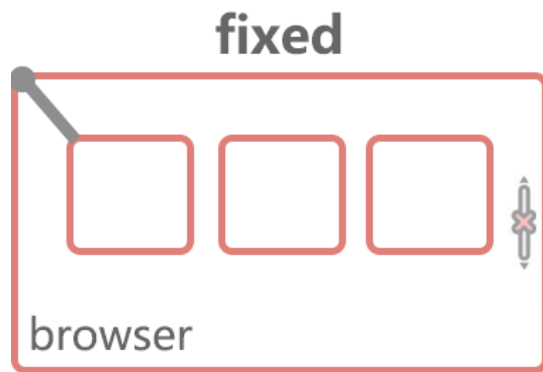
absolute(绝对定位):

- 表示, 相对于上级元素 (一般是父元素) 进行偏移, 即定位基点是父元素。
- 它有一个重要的限制条件: 定位基点 (一般是父元素) 不能是static定位, 否则定位基点就会变成整个网页的根元素html。
- 另外, absolute定位也必须搭配top、bottom、left、right这四个属性一起使用。
- 注意, absolute定位的元素会被"正常页面流"忽略, 即在"正常页面流"中, 该元素所占空间为零, 周边元素不受影响。



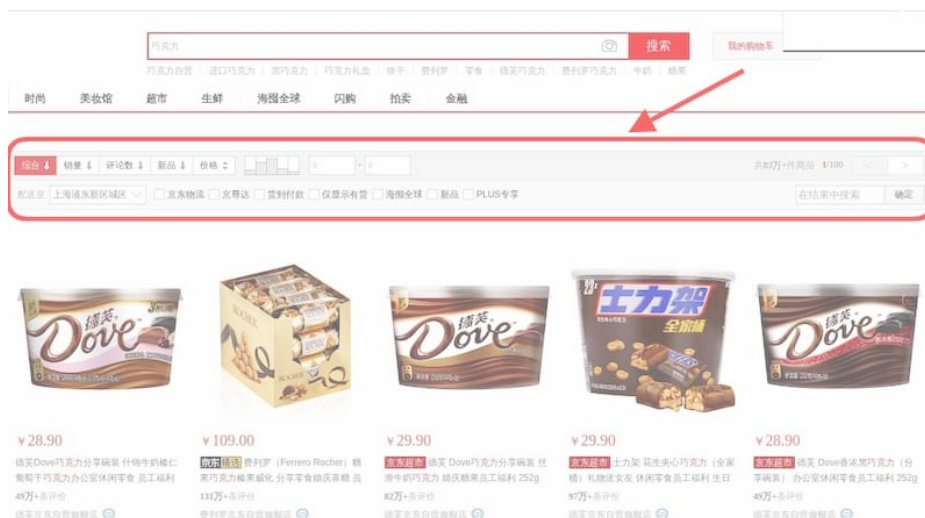
fixed(固定定位):

- 表示, 相对于视口 (viewport, 浏览器窗口) 进行偏移, 即定位基点是浏览器窗口。这会导致元素的位置不随页面滚动而变化, 好像固定在网页上一样。
- 它如果搭配top、bottom、left、right这四个属性一起使用, 表示元素的初始位置是基于视口计算的, 否则初始位置就是元素的默认位置。



sticky(粘连定位):

- 跟前面四个属性值都不一样，它会产生动态效果，很像relative和fixed的结合：一些时候是relative定位（定位基点是自身默认位置），另一些时候自动变成fixed定位（定位基点是视口）。
- 因此，它能够形成"动态固定"的效果。比如，网页的搜索工具栏，初始加载时在自己的默认位置（relative定位）。
- sticky生效的前提是，必须搭配top、bottom、left、right这四个属性一起使用，不能省略，否则等同于relative定位，不产生"动态固定"的效果。原因是这四个属性用来定义"偏移距离"，浏览器把它当作sticky的生效门槛。它的具体规则是，当页面滚动，父元素开始脱离视口时（即部分不可见），只要与sticky元素的距离达到生效门槛，relative定位自动切换为fixed定位；等到父元素完全脱离视口时（即完全不可见），fixed定位自动切换回relative定位；
- 页面向下滚动时，工具栏变成固定位置，始终停留在页面头部（fixed定位）
- 等到页面重新向上滚动回到原位，工具栏也会回到默认位置。





- sticky生效的前提是，必须搭配top、bottom、left、right这四个属性一起使用，不能省略，否则等同于relative定位，不产生"动态固定"的效果。原因是这四个属性用来定义"偏移距离"，浏览器把它当作sticky的生效门槛。
- 它的具体规则是，当页面滚动，父元素开始脱离视口时（即部分不可见），只要与sticky元素的距离达到生效门槛，relative定位自动切换为fixed定位；等到父元素完全脱离视口时（即完全不可见），fixed定位自动切换回relative定位。
- 除了已被淘汰的IE 以外，其他浏览器目前都支持sticky。但是，Safari 浏览器需要加上浏览器前缀-webkit-；
- position: -webkit-sticky; /* safari 浏览器 */ position: sticky; /* 其他浏览器 */

position fixed和sticky

<https://www.zhangxinxu.com/wordpress/2020/03/position-sticky-rules/>

position:sticky粘性定位与position:fixed固定定位；都可以在拖动滚动条的时候，将元素固定于指定位置，但是两者的区别也是非常巨大的；

position:fixed

- 元素可以被固定于页面指定位置，始终固定于此位置。
- 定位位置可以通过top、bottom、left与right属性设置。
- 如果属性同时设置，那么top的优先级高于bottom，left的优先级高于right。
- top、bottom、left与right无需显式设置，fixed定位也会生效。
- 元素的定位参考对象是当前窗口。
- 元素脱离文档流。
- 当前所有主流浏览器都支持此定位方式（不再考虑IE6浏览器）

position:sticky

- 元素可以被固定于页面指定位置，但并不一定始终固定于此位置。
- 定位位置可以通过top、bottom、left与right属性设置，但是此位置是一个临界值，也就是说元素只有达到设置的这个临界值才会固定，其他位置并不会固定。
- 如果属性同时设置，那么top的优先级高于bottom，left的优先级高于right。
- 必须至少显式设置top、bottom、left与right其中某一个属性值，sticky定位才会生效。
- 元素的定位参考对象距离其最近的overflow属性值为visible的具有滚动条的祖先元素，如果是以body或者body的父辈元素为考，那么定位参考对象是窗口。

- 元素不会脱离文档流。
- 浏览器兼容当前有点差，不过以后肯定会越来越好。Safari目前还需要-webkit-私有前缀。

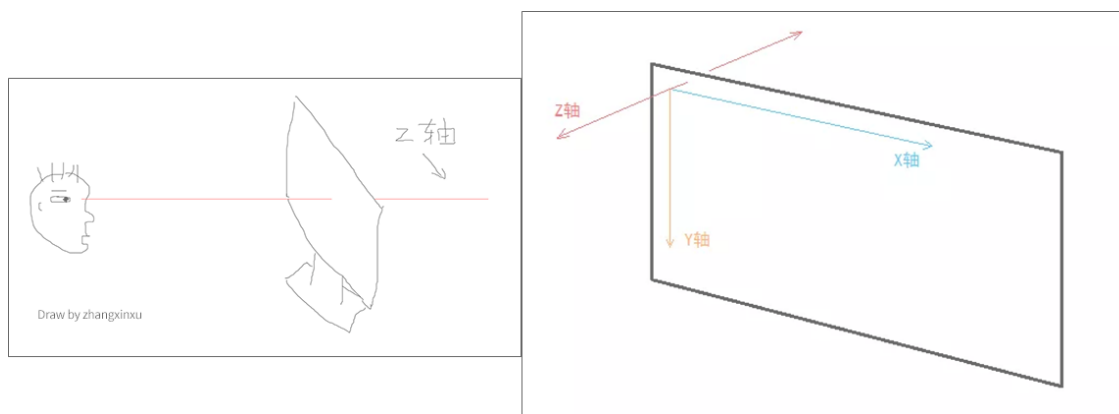
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *
			4-22			
		2-25	23-36			
		26-31	37-51	3.1-6	10-38	3.2-5.1
	12-15	32-58	52-55	6.1-7	39-41	6-7.1
6-10	16-17	59-63	56-70	7.1-11.1	42-56	8-11.4
11	18	64	71	12	57	12.1
		65-66	72-74	TP		

层叠上下文和层叠顺序

<https://www.zhangxinxu.com/wordpress/2016/01/understand-css-stacking-context-order-z-index/>

层叠上下文含义:

- 层叠上下文(stacking context), 是HTML中一个三维的概念。在CSS2.1规范中, 每个盒模型的位置是三维的, 分别是平面画布上的X轴, Y轴以及表示层叠的Z轴。一般情况下, 元素在页面上沿X轴Y轴平铺, 我们察觉不到它们在Z轴上的层叠关系。而一旦元素发生堆叠, 这时就能发现某个元素可能覆盖了另一个元素或者被另一个元素覆盖。
- 如果一个元素含有层叠上下文, (也就是说它是层叠上下文元素), 我们可以理解为这个元素在Z轴上就“高人一等”, 最终表现就是它离屏幕观察者更近。



层叠等级含义:

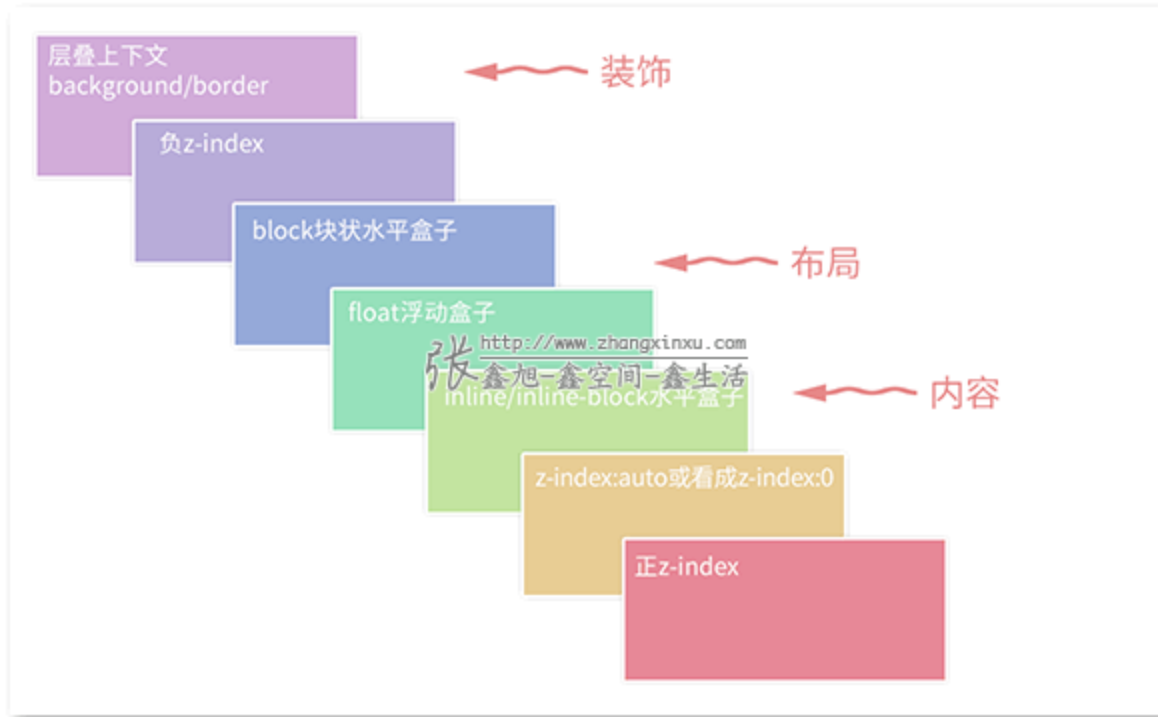
- 层叠等级(stacking level, 叫“层叠级别”/“层叠水平”也行)
- 在同一个层叠上下文中, 它描述定义的是该层叠上下文中的层叠上下文元素在Z轴上的上下顺序。
- 在其他普通元素中, 它描述定义的是这些普通元素在Z轴上的上下顺序;

总结:

- 普通元素的层叠等级优先由其所在的层叠上下文决定。
- 层叠等级的比较只有在当前层叠上下文元素中才有意义。不同层叠上下文中比较层叠等级是没有意义的。

层叠顺序:

"层叠顺序" (stacking order)表示元素发生层叠时按照特定的顺序规则在Z轴上垂直显示。由此可见，前面所说的“层叠上下文”和“层叠等级”是一种概念，而这里的“层叠顺序”是一种规则。



同一个堆叠上下文元素在Z轴的排列:

- 创建推叠上下文的元素的背景和边框
- 堆叠级别(z-index,stack level)为负值的堆叠上下文
- 常规流非定位的块盒
- 非定位的浮动盒子
- 常规流非定位行盒
- 任何z-index是auto的定位子元素，以及z-index是0的堆叠上下文
- 堆叠等级为正值的堆叠上下文

通俗理解:

- 当两个元素类型一样时，默认情况下后一个元素层级比前一个元素层级高。
- 在没有设置背景的情况下，元素的背景是透明的，并且允许后面的元素透上来。
- 块元素和其他任意除定位元素以外，文字层级比背景层级高。
- 浮动和块元素，浮动层级高。
- 浮动和行内块，行内块层级高。
- 浮动和行内，行内层级高。

- 定位和定位，后一个元素层级高。
- 定位比所有元素层级高。

背景和边框 < 小于0的z-index < 块 < 浮动 < 行内块 < 行内 < 定位 < 大于0的z-index;

固定定位 (position: fixed) 的特殊性

- 单纯是绝对/相对定位元素是无法创建一个层叠上下文的，需要同时满足 z-index 值不为 auto 的条件。然而，固定定位元素就不需要满足这个条件，显得尤为特殊。
- 固定定位元素无需满足 z-index 值不为 auto 的条件就可以创建层叠上下文；同样地，设置 z-index: auto 并不能撤销固定定位元素所创建的层叠上下文。
- 正常情况下，固定定位是相对于浏览器视窗 (viewport) 进行定位的，但是当其祖先元素中存在符合以下任意一个条件的元素时，固定定位元素会相对于该元素进行定位：
 - 1、transform 属性值不为 none；
 - 2、transform-style: preserve-3d；
 - 3、perspective 属性值不为 none；
 - 4、will-change 属性 指定了上面 3 个 CSS 属性中的任意一个；
- 然而，这种表现在不同的浏览器中有差异，譬如在 Safari 中只有上述第 1 点会对固定定位元素产生影响。

建议：通过对层叠上下文和层叠顺序的了解，我们知道，要控制元素间的层叠关系除了使用 z-index 属性外还有很多途径，而且比使用 z-index 要优雅得多。滥用 z-index 往往只会把层叠关系复杂化，造成代码难以维护。

层叠准则：

- 谁大谁上：当具有明显的层叠水平标示的时候，如识别的z-index值，在同一个层叠上下文领域，层叠水平值大的那一个覆盖小的那一个。通俗讲就是官大的压死官小的。
- 后来居上：当元素的层叠水平一致、层叠顺序相同的时候，在DOM流中处于后面的元素会覆盖前面的元素。

层叠上下文形成条件

[https://developer.cdn.mozilla.net/zh-](https://developer.cdn.mozilla.net/zh-CN/docs/Web/Guide/CSS/Understanding_z_index/The_stacking_context)

[CN/docs/Web/Guide/CSS/Understanding_z_index/The_stacking_context](https://developer.cdn.mozilla.net/zh-CN/docs/Web/Guide/CSS/Understanding_z_index/The_stacking_context)

层叠上下文(stacking context)，它是一块区域，这块区域由某个元素创建，它规定了该区域中的内容

在Z轴上排列的先后顺序。

凡是拥有层叠上下文的元素，将离用户最近，也就是越靠在Z轴前面。默认情况下只有根元素HTML会产生一个层叠上下文，并且元素一旦使用了一些属性也将会产生一个层叠上下文，如我们常用的定位属性。如两个层叠上下文相遇时，总是后一个层叠前一个，除非使用z-index来改变；

文档中的层叠上下文由满足以下任意一个条件的元素形成：

- 文档根元素 (<html>)
- position 值为 absolute (绝对定位) 或 relative (相对定位) 且 z-index 值不为 auto 的元素;
- position 值为 fixed (固定定位) 或 sticky (粘滞定位) 的元素 (粘滞定位适配所有移动设备上的浏览器, 但老的桌面浏览器不支持);
- flex (flexbox) 容器的子元素, 且 z-index 值不为 auto;
- grid (grid) 容器的子元素, 且 z-index 值不为 auto;
- opacity 属性值小于 1 的元素 (参见 the specification for opacity) ;
- mix-blend-mode 属性值不为 normal 的元素;
- 以下任意属性值不为 none 的元素:
 - transform
 - filter
 - perspective
 - clip-path
 - mask / mask-image / mask-border
- isolation 属性值为 isolate 的元素;
- -webkit-overflow-scrolling 属性值为 touch 的元素;
- will-change 值设定了任一属性而该属性在 non-initial 值时会创建层叠上下文的元素;
- contain 属性值为 layout、paint 或包含它们其中之一的合成值 (比如 contain: strict、contain: content) 的元素。

在层叠上下文中, 子元素同样也按照上面解释的规则进行层叠。重要的是, 其子级层叠上下文的 z-index 值只在父级中才有意义。子级层叠上下文被自动视为父级层叠上下文的一个独立单元。

总结:

- 层叠上下文可以包含在其他层叠上下文中, 并且一起创建一个层叠上下文的层级。
- 每个层叠上下文都完全独立于它的兄弟元素: 当处理层叠时只考虑子元素。
- 每个层叠上下文都是自包含的: 当一个元素的内容发生层叠后, 该元素将被作为整体在父级层叠上下文中按顺序进行层叠。

层叠上下文对z-index的影响

含义: z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。该属性设置一个定位元素沿 z 轴的位置, z 轴定义为垂直延伸到显示区的轴。如果为正数, 则离用户更近, 为负数则表示离用户更远

- 层叠上下文 (stacking context) 并不只是 z-index (必须配合 position 才能生效) 才能创建, 还有很多其他元素 (如: opacity、transform 等) 也可以创建层叠上下文;
- 在存在层叠上下文的情况下, z-index 的大小决定了层叠水平 (stacking level), 即谁在谁上面, 这是“谁大谁上”原则;
- 层叠水平的比较只有在同一级别的 DOM 节点的层叠上下文中才有意义;

- 在同一DOM节点，并且层级水平一样的情况下，在HTML文档中写在后面的元素会遮住前面的元素（后者会在前者上面），这是“后来居上”原则；

