

Dismiss

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Sign up

fixed 定位失效 | 不受控制的 position:fixed #24

Open

chokcoco opened this issue on 14 Aug 2017 · 4 comments

New issue

chokcoco commented on 14 Aug 2017 • edited

Owner

本文为纯理论文章，可能会略微枯燥。

大家都知道，`position:fixed` 在日常的页面布局中非常常用，在许多布局中起到了关键的作用。它的作用是：

`position:fixed` 的元素将相对于屏幕视口 (viewport) 的位置来指定其位置。并且元素的位置在屏幕滚动时不会改变。

但是，在许多特定的场合，指定了 `position:fixed` 的元素却无法相对于屏幕视口进行定位。这是为何呢？

失效的 `position:fixed`

在许多情况下，`position:fixed` 将会失效。MDN 用一句话概括了这种情况：

- 当元素祖先的 `transform` 属性非 `none` 时，容器由视口改为该祖先。

What！还有这种操作？可能有部分同学还没 get 到上面这句话的意思，通俗的讲就是指定了 `position:fixed` 的元素，如果其祖先元素存在非 `none` 的 `transform` 值，那么该元素将相对于设定了 `transform` 的祖先元素进行定位。

那么，为什么会发生这种情况呢？说好的相对视口 (Viewport) 定位呢？

这个问题，就牵涉到了 `Stacking Context`，也就是堆叠上下文的概念了。解释上面的问题分为两步：

- 任何非 `none` 的 `transform` 值都会导致一个堆叠上下文 (`Stacking Context`) 和包含块 (`Containing Block`) 的创建。
- 由于堆叠上下文的创建，该元素会影响其子元素的固定定位。设置了 `position:fixed` 的子元素将不会基于 `viewport` 定位，而是基于这个父元素。

Stacking Context -- 堆叠上下文

好的嘛，好的嘛，又冒出新的名词了，堆叠上下文 (又译作层叠上下文)，又是什么？

堆叠上下文 (`Stacking Context`)：堆叠上下文是 HTML 元素的三维概念，这些 HTML 元素在一条假想的相对于面向 (电脑屏幕的) 视窗或者网页的用户的 `z` 轴上延伸，HTML 元素依据其自身属性按照优先级顺序占用层叠上下文的空间。

概念比较抽象，简单理解，记住 **生成了 `Stacking Context` 的元素会影响该元素的层叠关系与定位关系。**

关于 **生成了 `Stacking Context` 的元素会影响该元素的层叠关系** 这一点，具体可以看看这篇文章 [层叠顺序 \(`stacking level` \) 与堆栈上下文 \(`stacking context` \) 知多少？](#)

而本文提到了**生成了 `Stacking Context` 的元素会影响该元素定位关系**。按照上面的说法，堆叠上下文的创建，该元素会影响其子元素的固定定位。设置了 `position:fixed` 的子元素将不会基于 `viewport` 定位，而是基于这个父元素。

Assignees

No one assigned

Labels

原理

Projects

None yet

Milestone

No milestone

4 participants

那么问题来了，是否所有能够生成堆叠上下文的元素，都会使得其子元素的 `position:fixed` 相对它，而不是相对视口（`Viewport`）进行定位呢？

创建堆叠上下文的方式

为此，首先要找到所有能够使元素生成堆叠上下文的方法。

So，如何触发一个元素形成 堆叠上下文 ？方法如下（参考自 [MDN](#)）：

- 根元素 (HTML),
- z-index 值不为 "auto"的 绝对/相对定位，
- 一个 z-index 值不为 "auto"的 flex 项目 (flex item)，即：父元素 `display: flex|inline-flex`，
- opacity 属性值小于 1 的元素（参考 the specification for opacity），
- transform 属性值不为 "none"的元素，
- mix-blend-mode 属性值不为 "normal"的元素，
- filter值不为"none"的元素，
- perspective值不为"none"的元素，
- isolation 属性被设置为 "isolate"的元素，
- position: fixed
- 在 will-change 中指定了任意 CSS 属性，即便你没有直接指定这些属性的值
- -webkit-overflow-scrolling 属性被设置 "touch"的元素

接下来，我们要验证，是否所有设置了上面属性样式之一的元素，都有使其子元素的 `position: fixed` 失效的能力？

为此我做了下面一个小实验，基于最新的 Blink 内核。可戳：

[层叠上下文对 fixed 定位的影响](#)（不同内核下表现可能不一致）



我们设置两个父子 `div`，子元素 `fixed` 定位，通过修改父元素生成层叠上下文，观察子元素的 `fixed` 定位是否不再相对视口。

```
<div class="container">
  <div class="fixed"> </div>
</div>
```

最初的 CSS：

```
.container {
  width:10vw;
  height: 10vw;
  background: rgba(255, 100, 100, .8);
}

.fixed {
  position: fixed;
  top: 1vw;
  left: 1vw;
  right: 1vw;
  bottom: 1vw;
  background: rgba(100, 100, 255, .8);
}
```

一探 position:fixed 失效的最终原因

通过上面的试验，在最新的 Blink 内核下，发现并不是所有能够生成层叠上下文的元素都会使得 `position:fixed` 失效，但也不止 `transform` 会使 `position:fixed` 失效。

所以，MDN 关于 `position:fixed` 的补充描述不够完善。下述 4 种方式目前都会使得 `position:fixed` 定位的基准元素改变（**本文重点**）：

1. transform 属性值不为 none 的元素

2. 设置了 `transform-style: preserve-3d` 的元素
3. `perspective` 值不为 `none` 的元素
4. 在 `will-change` 中指定了任意 CSS 属性

不同内核的不同表现

完了吗？没有！我们再看看其他内核下的表现。



上面也谈到了，上述结论是在最新的 Chrome 浏览器下（Blink内核），经过测试发现，在 MAC 下的 Safari 浏览器（WebKit内核，Version 9.1.2 (11601.7.7)）和 IE Trident/ 内核及 Edge 浏览器下，上述三种方式都不会改变 `position: fixed` 的表现！

所以，当遇到 `position: fixed` 定位基准元素改变的时候，需要具体问题具体分析，多尝试一下，根据需要兼容适配的浏览器作出调整，不能一概而论。

position: fixed 的其他问题

当然，`position: fixed` 在移动端实现头部、底部模块定位。或者是在 `position: fixed` 中使用了 `input` 也会存在一些问题，这个有很多文章都描述过并且存在很多解决方案，本文不讨论这块问题。

这方面的问题，可以看看这篇文章：[移动端web页面使用position:fixed问题总结](#)



sgyy1994 commented on 25 Sep 2017

作者的解答太棒了，刚刚在业务中遇到了这个问题，感谢~



👤 chokcoco changed the title from 不受控制的 `position:fixed` to fixed定位失效 | 不受控制的 `position:fixed` on 25 Sep 2017



👤 chokcoco changed the title from fixed定位失效 | 不受控制的 `position:fixed` to fixed 定位失效 | 不受控制的 `position:fixed` on 25 Sep 2017



👤 chokcoco added the 原理 label on 26 Sep 2017



👤 hnsxxscyx referenced this issue in amfe/lib-flexible on 28 Sep 2017

fixed 定位变成了相对于 body定位，而不是相对于浏览器可视区定位 #92

Closed



liuyiliuyi commented on 3 Oct 2017 • edited by chokcoco ▾

使用`transform-style: preserve-3d`属性也会使 `position:fixed` 的元素相对于该祖先元素定位



deepkolos commented on 17 Oct 2017

我遇过在使用`transform` 动画的时候 `position:fixed`，如果有`top/bottom`的时候会失效 `left/right`则不会的情况
[探究position:fixed在css动画过程中的行为~](#)



chokcoco commented on 19 Oct 2017

Owner

@liuyiliuyi

3Q，我试了一下 设置了 `transform` 属性值不为 `"none"`的元素确实也会。在想是否这条可以归类为 `transform 属性值不为 none 的元素` 这里。