

移动端疑难杂症：1px 问题如何解决

1px 问题的起因

1px 问题指的是：在一些 `Retina` 屏幕 的机型上，移动端页面的 1px 会变得很粗，呈现出不止 1px 的效果。

原因很简单——CSS 中的 1px 并不能和移动设备上的 1px 划等号。它们之间的比例关系有一个专门的属性来描述：

```
1 window.devicePixelRatio = 设备的物理像素 / CSS像素。
```

```
1 devicePixelRatio: 2
```

意味着我设置的 1px CSS 像素，在这个设备上实际会用 2 个物理像素单元来进行渲染，所以实际看到的一定会比 1px 粗一些

直接写 0.5px

```
1 <div id="container" data-device={{window.devicePixelRatio}}></div>
2
3 #container[data-device="2"] {
4   border:0.5px solid #333
5 }
```

先放大后缩小

4条边框

```
1 #container::after{
2   position:absolute;
3   top: 0;
4   left: 0;
5   width: 200%;
6   height: 200%;
7   content:"";
8   transform: scale(0.5);
9   transform-origin: left top;
10  box-sizing: border-box;
11  border: 1px solid #333;
```

```
12 }  
13  
14
```

1条边框

```
1  ::after{  
2      position: absolute;  
3      content: '';  
4      background-color: #e5e5e5;  
5      display: block;  
6      width: 100%;  
7      height: 1px; /*no*/  
8      transform: scale(1, 0.5);  
9      top: 0;  
10     left: 0;  
11 }  
12
```

viewport 缩放来解决

```
1  const scale = 1 / window.devicePixelRatio;  
2  // 这里 metaEl 指的是 meta 标签对应的 Dom  
3  
4  metaEl.setAttribute('content', width=device-width,user-scalable=no,initial-  
    scale=${scale},maximum-scale=${scale},minimum-scale=${scale});
```

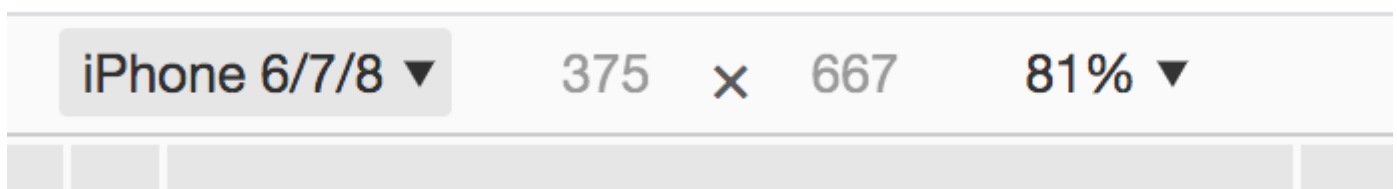
但这样做的副作用也很大，整个页面被缩放了。这时候我们的 1px 已经被处理成物理像素大小了，这样的大小在手机上显示边框很合适。但仔细想想，一些原本不需要被缩小的内容，比如文字、图片等，也被无差别缩小掉了。

没错，页面上的图片和文字都被缩小了 dpr 倍，

需要把原来写的 css 像素都再乘以 dpr 倍。这样才能既解决 1px 问题，又与原来大小保持一致头疼。

难道自己把每个 css 属性乘以 dpr 倍吗？

哈哈 设计师上场了：



视觉稿一开始就以设备物理像素来给出（设计师直接给我750的设计稿，因为 dpr 为 2），那么我们就不要再乘以 dpr 倍了，直接设计稿里 px 多少，css 里 px 就写多少即可。

那如果设计稿和设备物理像素不一样怎么办？

比如设计稿是 750 的，手机 **screen.width = 375 dpr = 3**，按照上面的公式，**viewport-width = 1125**，那么按照设计稿来写 px，虽然高清了但肯定是铺不满的，还需要再适配一下。

这个适配问题就可以用 rem 来解决

针对不同手机屏幕尺寸和dpr动态改变根节点html的 font-size 大小

```
1 rem = _baseFontSize / _psdWidth * docEl.clientWidth;
```

在这个公式里，假如视觉规范是 750（_psdWidth = 750），那么我们就确保了 rem 无论在任何页面里，都一定是占据了页面的 100/750 这个比例，也就是说 1px = 0.01rem 永远占据这个页面的 1/750