

# 基于screen.width的伪响应式开发

这篇文章发布于 2016年06月3日, 星期五, 23:09, 归类于 Web综合。 阅读 22655 次, 今日 8 次 16 条评论

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5416>

## 一、站在用户的角度看问题

一个用户, 访问一个web页面的真实场景是怎样的呢?

下面是某用户访问某站点的一个场景:

1. 小明打开了自己的电脑, 访问了鑫空间-鑫生活;
2. 小明体内洪荒之力无法控制, 疯狂拖动浏览器改变其宽度感受页面布局变化;
3. 卧槽, 发现页面居然是响应式的, 不由得感叹: 实现的够骚气! 🍊



很显然, 小明要么是雷蜜胡粉心碎抓狂, 要不就是前端开发职业病发作难耐。作为真实的用户是不会把浏览器刻意缩小去体验的, 你想想看你平时上网的时候, 会把浏览器窗口拉到很小吗? 怕是嫌屏幕小, 恨不得显示器大到铺满办公桌吧~~

## 二、回到传统的响应式开发

传统的响应式实现往往基于基于media query查询, 例如:

```
@media screen and (max-width: 480px) {  
    /* 窄屏下 */  
}
```

这是基于CSS的布局控制, 因此, 当我们缩小浏览器窗口, 可以即时看到布局变化。但是, 这种实现在我看来, 除了让总监大人可以方便体验窄屏效果外, 就然并卵了! 而反倒是有可能增加了额外的资源消耗和开发成本。😞

@media 可以即时控制宽窄布局, 很自然地, 我们的JS逻辑也要一并跟上。假如说, PC和mobile有很多不同的交互逻辑, 我们的HTML是同一套, 当我们在响应PC和Mobile零界点不停变换的时候, 我们的JS逻辑是不是也要跟着即时变化!

这就导致问题来了, CSS浏览器渲染, 本身即时响应。但JS且不一样, 我们必须实时监测是PC宽度还是

Mobile宽度，同时PC的click事件和Mobile的touch事件可能就在同一个元素上搞基了，也蛮累心的。为了我们自己省心，我们就可能去限制设计师再做响应式设计的时候，两者差异不要太大。我去，技术已经不是帮助产品设计体验升级，而是去制约设计了，贵司的设计师好惨！

还有一个问题就是资源消耗的问题，拿网站头图举例，PC的头图可能是张大大的长图，Mobile是个方方的图片。即时响应也就意味着这两个图都可能被加载。

那有没有什么办法既能满足响应式的需求，同时我们开发这边不要那么烦心呢？

试试使用 `screen.width` 来做伪响应式开发。

### 三、`screen.width`伪响应式开发

首先要了解下不会说谎的 `screen.width`，`screen.width` 顾名思义就是屏幕的宽度，对，屏幕的宽度，与显示器宽度没有任何关系，就算你把显示器宽度缩小到芝麻糊那么大，`screen.width` 还是不会变。

在“[CSSOM视图模式相关整理](#)”一文中有过介绍，包括IE6在内的浏览器全支持。

由于 `screen.width` 不会说谎，我们就可以瞬间确定用户实现的宽屏设备还是窄屏设备，而 `@media screen` 的宽度是浏览器的可用宽度，很容易就被用户欺骗的。

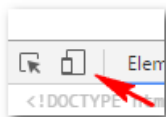
关于`screen.width`可能的疑问

#### 1. 手机如果横着访问会怎样？

根据我在自己手机上的测试，你手机横过来还是竖过来，`screen.width` 都是你屏幕竖着浏览时候的宽度。好比肾6，你横竖浏览，此时 `screen.width` 都是 375px；

#### 2. PC浏览器如何测试？

测试的话不是缩小浏览器宽度，而是打开控制台，进入手机模式，此时，`screen.width` 也会跟着一起变哦~记得刷新下页面~

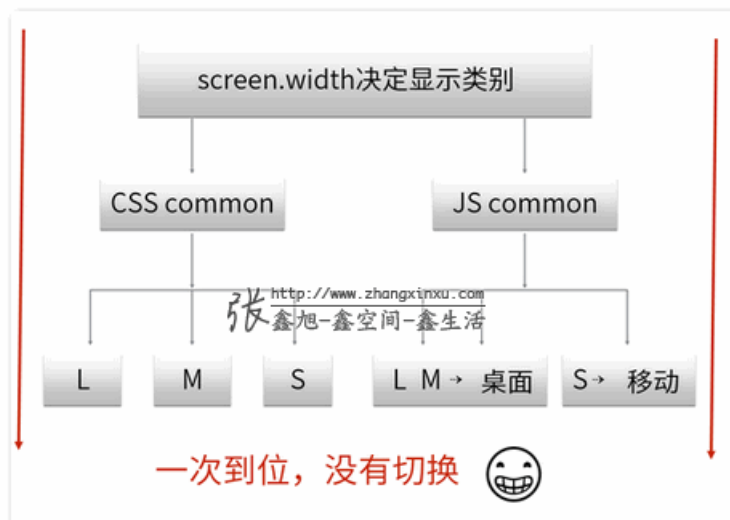


只要我们确认了用户的屏幕尺寸，我们就可以在一开始就确定我们的页面布局以及所需要的交互，例如，可以在 `<head>` 标签内放入这么一段内联script:

```
(function(doc, win) {
    var screenWidth = 0, size = 'M', root = doc.documentElement;
    if (window.screen && screen.width) {
        screenWidth = screen.width;
        if (screenWidth > 1920) {
            // 超大屏, 例如iMac
            size = 'L';
        } else if (screenWidth < 480) {
            // 小屏, 如手机
            size = 'S';
        }
    }
    // 标记CSS
    root.className = size;
    // 标记JS
    win.SIZE = size;
})(document, window);
```

上面的脚本在页面加载的一开始，就确定了是大屏，普通屏还是小屏，然后再执行响应的渲染和脚本执行。您可以根据自己实际项目，修改上面的 `size` 变量。

于是乎，我们无论是CSS渲染还是JS逻辑处理，都是1条线下来，完全没有 `@media screen` 即时切换而不得已耦合在一起的JS逻辑处理。



典型的伪响应式代码如下：

```
.S .example {
  /* 移动端的样式 */
}
```

```
if (window.SIZE == 'S') {
  // 移动端的处理
} else {
  // 桌面端的处理
}
```

考虑到真实的使用场景，基于 `screen.width` 的伪响应式开发对用户而言，没有任何区别，该什么设备看到的还是那个设备该有的样子；但是，对于开发人员，也就是我们前端开发自己而言，那就大不一样了，一条故事线下来，逻辑清晰，处理轻松，设计师把PC和Mobile涉及的差异再明显，我也能从容应对，对吧，`if{} else{}` 里面互不干扰，好轻松~ 😊

## 四、结语

本文并不是否决基于media queries的响应式处理，只是提供另外一个解决问题的思路。如果你的PC和Mobile的有很多不同的逻辑处理，试试这种一棒子打死的“响应式”策略。

然后，本文的策略是经过真实目前在线的有一定分量的项目验证过的，同事认同，Boss认同（除了缩小屏幕没法体验手机），自己跳开了很多坑，更加认同。有机会，你也不妨试试。



《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« canvas图形绘制之星空、噪点与烟雾效果

基于用户行为的图片等资源预加载 »

#### 猜你喜欢

- CSSOM视图模式(CSSOM View Module)相关整理
- 基于vw等viewport视区单位配合rem响应式排版和布局
- CSS3 Media Queries的些野史外传
- 翻译 - CSS高峰会议内容精选
- 应运而生的web页面响应布局
- 翻译: web响应设计, 乏味!
- 热门: 响应图片(Responsive Images)技术简介
- 伪类+js实现CSS3 media queries跨界准确判断
- 博闻强识: 了解CSS中的@ AT规则

分享到: 1

标签: media queries, screenX, screenY, window.screen, 响应式布局

#### 发表评论 (目前16条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

#### 1. 可妮兔说道:

2018年04月12日 17:20

大神。我想问一下,最后面那个

```
if (window.SIZE === 'S') {
```

```
// 移动端的处理
```

```
} else {
```

```
// 桌面端的处理
```

```
}
```

是写在哪里? 没怎么看懂、百度了也没查到有关于你写的之外的这个的理解。

能不能再详细解释一下这个? 我也觉得媒体查询有点麻烦。你这个方法很好。可是我没太懂

[回复](#)



可妮兔说道：  
2018年04月12日 17:25

大神求救

[回复](#)



张 鑫旭说道：  
2018年04月13日 10:50

写在<head>中~

[回复](#)



可妮兔说道：  
2018年04月20日 08:58

这个写在css里面写  
.S.example {  
/\* 移动端的样式 \*/  
}  
这个。  
下面这个写在head里面？

{大神。我收到了你的书。我会好好看的！}

```
(function(doc, win) {  
  var screenWidth = 0, size = 'M', root = doc.documentElement;  
  if (window.screen && screen.width) {  
    screenWidth = screen.width;  
    if (screenWidth > 1920) {  
      // 超大屏，例如iMac  
      size = 'L';  
    } else if (screenWidth < 480) {  
      // 小屏，如手机  
      size = 'S';  
    }  
  }  
  // 标记CSS  
  root.className = size;  
  // 标记JS  
  win.SIZE = size;  
})(document, window);
```

[回复](#)



2. 芹菜说道：  
2016年12月13日 14:32

三星手机screen.width 不准确，三星note3获取宽度为1080，这个怎么弄呀

[回复](#)



3. 子健说道：  
2016年09月27日 16:39

有些同志，用浏览器看网站的时候，浏览器并不全屏，那就悲剧了

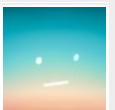
[回复](#)



riophae说道：  
2017年02月6日 22:29

和浏览器窗口大小没关系，它返回的值只和屏幕分辨率有关系

[回复](#)



4. 阿威说道：  
2016年09月26日 17:12

uc浏览器上并不能识别这一招啊，是怎么处理uc的？



[回复](#)

5. **横屏竖屏不一样**说道:  
2016年08月30日 18:06

旭哥，用谷歌浏览器，手机模式下，横屏竖屏不一样。

[回复](#)



6. **wingmeng**说道:  
2016年06月16日 15:32

这种思想非常赞，让JS与CSS整齐划一，整个世界都清爽了。不过我在测试时发现 screen.width并不能获取移动设备的真实屏幕宽度，跑Android机，开weinre，控制台用 screen.width 得到的是手机分辨率，不是真实的屏幕宽度，样式也没有应用到。是 screen.width 这货不靠谱还是我哪里搞错了？

[回复](#)



蛤蛤说道:  
2016年07月18日 16:45

少了viewport

[回复](#)



7. **路人女主**说道:  
2016年06月14日 13:08

这个功能好棒。。

[回复](#)



8. **DeathGhost**说道:  
2016年06月13日 01:08

呵呵,嗯，不错，学习了....  
这个我现在一般就处理PC和手机用css3处理...  
`@media (min/max-width:value){...}`  
个人站随便弄弄，凡是小于1200px就手机模式显示了...哈哈  
你的站点不错，搜索基本跑你这里来了...  
赞~

[回复](#)



9. **wingmeng**说道:  
2016年06月7日 17:48

好文！实现了js和css的统一协作

[回复](#)



10. **crli**说道:  
2016年06月6日 10:35

真在处理这种问题。。。用上了，真好

[回复](#)



11. **dolymood**说道:  
2016年06月4日 23:17

比较痛苦的是有些手机会有坑 你取到的值是真实的分辨率宽 特殊判断？

[回复](#)



## 最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果
- » 从天猫某活动视频不必要的3次请求说起
- » CSS vector-effect与SVG stroke描边缩放
- » CSS ::backdrop伪元素是干嘛用的?
- » 周知: CSS -webkit-伪元素选择器不再导致整行无效

## 今日热门

- » 常见的CSS图形绘制合集 <sup>(190)</sup>
- » 未来必热: SVG Sprite技术介绍 <sup>(119)</sup>
- » 粉丝群第1期CSS小测点评与答疑 <sup>(115)</sup>
- » HTML5终极备忘大全 (图片版+文字版) <sup>(93)</sup>
- » 让所有浏览器支持HTML5 video视频标签 <sup>(86)</sup>
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 <sup>(82)</sup>
- » CSS3下的147个颜色名称及对应颜色值 <sup>(79)</sup>
- » 视区相关单位vw, vh..简介以及可实际应用场景 <sup>(76)</sup>
- » 写给自己看的display: flex布局教程 <sup>(76)</sup>
- » 小tips: 纯CSS实现打字动画效果 <sup>(76)</sup>

## 今年热议

- » 《CSS世界》女主角诚寻靠谱一起奋斗之人 <sup>(76)</sup>
- » 不借助Echarts等图形框架原生JS快速实现折线图效果 <sup>(64)</sup>
- » 看, for..in和for..of在那里吵架! <sup>(60)</sup>
- » 是时候好好安利下LuLu UI框架了! <sup>(47)</sup>
- » 原来浏览器原生支持JS Base64编码解码 <sup>(35)</sup>
- » 妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现 <sup>(33)</sup>
- » 炫酷H5中序列图片视频化播放的高性能实现 <sup>(31)</sup>
- » CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 <sup>(30)</sup>
- » windows系统下批量删除OS X系统.DS\_Store文件 <sup>(26)</sup>
- » 写给自己看的display: flex布局教程 <sup>(26)</sup>

## 猜你喜欢

- CSSOM视图模式(CSSOM View Module)相关整理
- 基于vw等viewport视区单位配合rem响应式排版和布局
- CSS3 Media Queries的些野史外传
- 翻译 - CSS高峰会议内容精选
- 应运而生的web页面响应布局
- 翻译: web响应设计, 乏味!
- 热门: 响应图片(Responsive Images)技术简介
- 伪类+js实现CSS3 media queries跨界准确判断

