**困扰多时的问题**

在这之前做web app开发的的时候，在自适应方面一般都是宽度通过百分比，高度以iPhone6跟iPhone5之间的一个平衡值写死，我们的设计稿都是iPhone5的640 \* 1136标准，所以高度一般取个大概值，各种图标的宽高也是取平衡值写死，然后部分样式通过媒体查询来设置，例如背景图的多倍图、基础字体大小、图标宽高。

这样做的弊端很明显：

1.做出来的页面在各种手机端看起来的物理大小（高度）是一样的，所以在大屏手机会觉得页面稍小，小屏手机页面稍大  
2.如果要使高度能更好的适应各种手机屏幕，需要写太多的媒体查询样式，效率低下  
3.全屏背景图片跟页面元素需要耦合时，元素位置的确定尤为困难（可能需要通过百分比去确定元素的横向位置，但始终会有误差）  
4….

最近在微博上看到流云诸葛总结的一篇文章《[从网易与淘宝的font-size思考前端设计稿与工作流](http://www.codeceo.com/article/font-size-web-design.html" \t "_blank)》，其中介绍到的几种web app适配方案，我们现在的做法恰好是跟拉勾网类似的简单方案，当然就会有上面我提到的一些问题，最后经过预研和demo测试，我们采取了网易跟淘宝的方案，其实这两者的方案是大同小异，都是基于rem的适配方案。

**解决问题的方案**

网易跟淘宝的方案介绍在上面流云诸葛的[文章](http://www.codeceo.com/article/font-size-web-design.html" \t "_blank)中已经写的很清楚了，建议可以先看看那篇文章再阅读下面我所说的，可能会更加清晰。

**方案的简单介绍: 基于rem**

前提：页面元素的布局尺寸全都以设计稿为基准等比例设置。

给html根节点设置一个基础font-size值，然后页面的所有元素布局均相对于该font-size值采用rem单位设定。那么基础的font-size值该如何取呢？

假如通过媒体查询设置font-size，只能解决一部分的情况，而且并不能完成适配，因为手机屏幕宽度类型实在太多了，所以font-size的取值要通过js计算，取当前viewport的deviceWidth与设计稿的宽 的 比例值，例如：我们的设计稿尺寸都是640px的，iphone5的deviceWidth是320px，那么计算出来的font-size值就是 320 / 640 = 0.5，因为得出的font-size太小，不方便计算，且有的浏览器可能不兼容太小字号，所以将font-size放大100倍，所以最终计算出来的font-size为 320 / 640 \* 100 = 50(px); 当然，这个值是根据设计稿来计算的，所以根据计算规则，下面列出几种常见设计稿相应的font-size值：

deviceWidth = 320，font-size = 320 / 6.4 = 50px  
deviceWidth = 375，font-size = 375 / 6.4 = 58.59375px  
deviceWidth = 414，font-size = 414 / 6.4 = 64.6875px  
deviceWidth = 500，font-size = 500 / 6.4 = 78.125px

可在script标签加上如下代码

|  |
| --- |
| (function () {      document.addEventListener('DOMContentLoaded', function () {          var html = document.documentElement;          var windowWidth = html.clientWidth;          html.style.fontSize = windowWidth / 6.4 + 'px';          // 等价于html.style.fontSize = windowWidth / 640 \* 100 + 'px';      }, false);  })();    // 这个6.4就是根据设计稿的横向宽度来确定的，假如你的设计稿是750  // 那么 html.style.fontSize = windowWidth / 7.5 + 'px'; |

至此，font-size的基础值就确定好了，而且知道该font-size值是手机deviceWidth跟设计稿的比例值 的 100倍（重要）

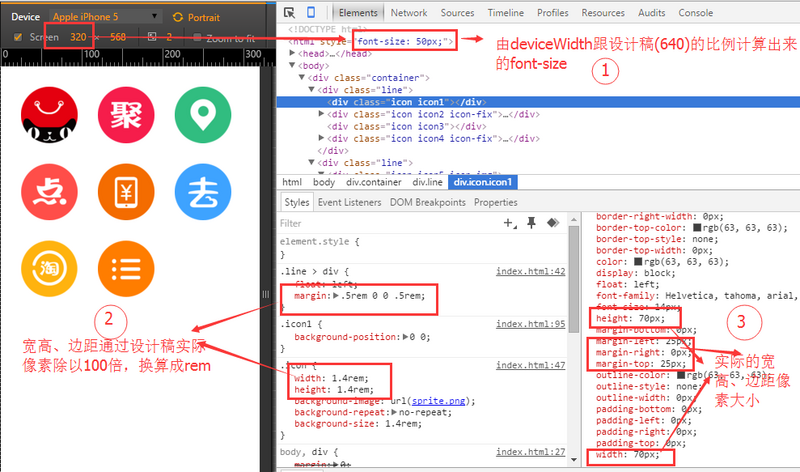
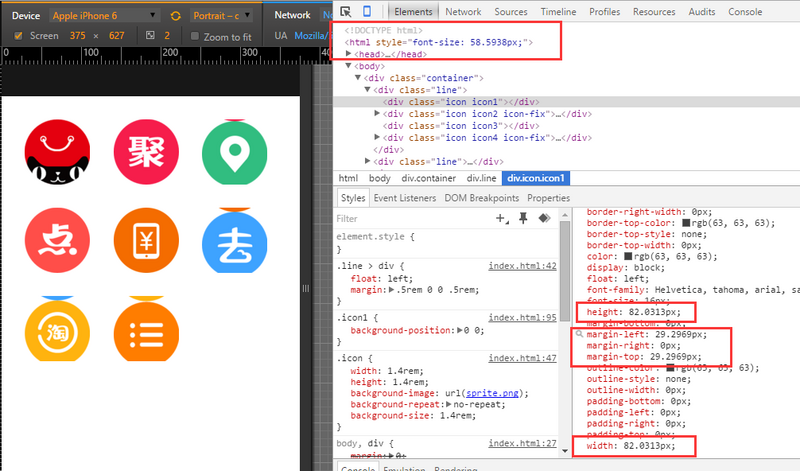
**那么页面元素该如何设置宽高、边距…**

例如：一个设计稿宽高为140px的图标，左边距为50px，那么它的css应该这样写

|  |
| --- |
| .icon {      width: 1.4rem; /\* 像素换算rem：140px / 100 = 1.4rem \*/      height: 1.4rem;      margin: 0 0 0 .5rem;  } |

因为html的font-size是放大了100倍，所以计算rem时，要用设计稿的实际像素除以100，140px / 100 = 1.4rem; 最后实际的像素大小就会由deviceWidth跟设计稿的横向宽 的 比例 自动计算出来。

如图iPhone5下面的效果：

  
iphone6的效果：  
  
可以看出来：html的font-size动态根据deviceWidth改变，图标的宽高、边距等也根据font-size动态按比例变化，大功告成了？不对，相信机智的你已经看到貌似在iPhone6的下有的图标背景错位了。。是的，这暴露出了一个背景使用雪碧图的一个弊端（由于font-size小数点太多，计算出实际背景图大小background-size跟背景图位置background-position时浏览器精度不够可能就会出现位置的偏差（我猜的），这个后面还会详细讲解决方案）

到这里，设置宽高、边距等都OK了，接下来…

**其他元素的字体大小该如何设置？**

在流云诸葛的文章中讲到，网易跟淘宝的做法都是使用额外的媒体查询设置几种字体大小，例如：

|  |
| --- |
| @media screen and (max-width: 320px) {      body{font-size: 14px;}  }  @media screen and (min-width: 321px) and (max-width: 413px) {      body{font-size: 16px;}  }  @media screen and (min-width: 414px) and (max-width: 639px) {      body{font-size: 17px;}  }  @media screen and (min-width: 640px) {      body{font-size: 18px;}  } |

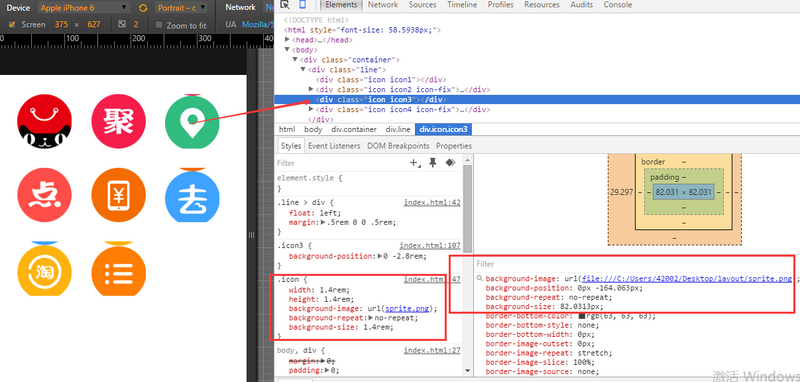
可为什么不用rem呢？后来去查了一番资料，发现有一种叫做点阵字体的存在（什么是点阵字体），也叫作位图字体，位图我们都知道，跟矢量图是有区别的，就是放大会模糊，所以点阵字体也是放大会模糊的，如果根据rem设置字体大小，字体会自由缩放，可能就会导致点阵字体模糊，所以需要设定使用几种固定大小的字体。不过，在正常情况下，系统自带的字体都是矢量字体，所以使用rem为单位是没有问题的，除非你的网页需要用到特殊的点阵字体。

总结：如果网页没有用到特殊的点阵字体，字体单位使用rem，如果用到了点阵字体，字体需要通过媒体查询设置几种固定大小的字体

**关于背景图片的错位问题**

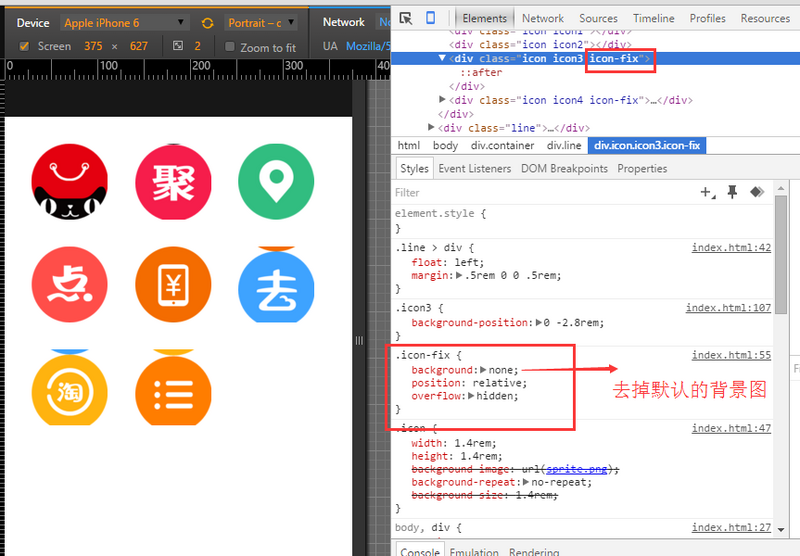
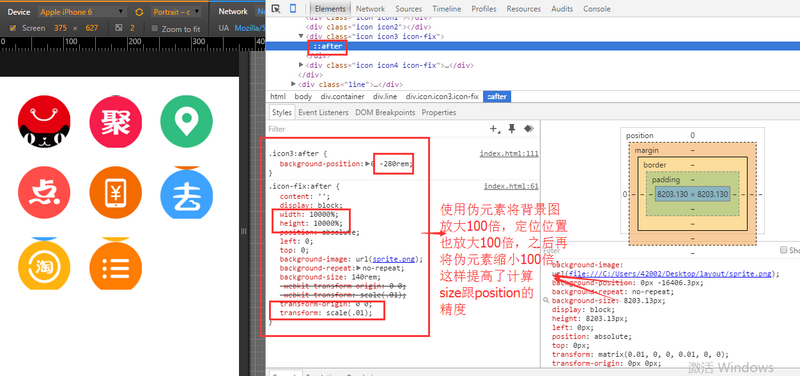
上面已经发现了，通过换算rem设置background-size跟background-position的时候，在一些手机型号下会出现背景图错位的情况，可是如果不用rem设置的话，又不能达到适配的目的。（background-size、background-position的rem换算方法跟前面讲的宽高设定一样，都是设计稿尺寸（这时应该是雪碧图的原始尺寸）除以100倍）

最后经过尝试，得出了几种解决方案：  
1、如图（推荐方案）：



图标的样式

|  |
| --- |
| .icon {      width: 1.4rem;      height: 1.4rem;      background-image: url(sprite.png);      background-repeat: no-repeat;      background-size: 1.4rem;  }    .icon3 {      background-position: 0 -2.8rem;  } |

解决方法，如图：  
  


所有相关代码 [传送门](https://github.com/DMQ/notes/tree/master/webApp-layout)

2、不使用雪碧图，使用单个背景图，这个时候就不存在background-position的需要，只需设定background-size: contain;即可，这样做的弊端就在于无法使用雪碧图，图片请求增多，适用于页面图标较少的情况

3、使用嵌套img标签，通过绝对定位模拟background-position，具体请看 [responsive-sprites](http://tobyj.net/responsive-sprites/)，这种做法需要更多的标签，且img图片只能放图标尺寸大小一样的雪碧图，而且不能通过媒体查询使用多倍图

以上3中解决方案第一种最优，当然有些特殊情况可能需要按需选择！

最后发现一篇研究rem产生小数点像素的问题的文章，感兴趣的可戳 《[rem 产生的小数像素问题](http://taobaofed.org/blog/2015/11/04/mobile-rem-problem/)》

**写在最后**

关于web app的探索之路还很长，以上纯粹个人在学习过程的一些探索和研究，肯定会有不足和错漏的地方，只有在不断的实践中去修正。如果大家发现其中有错或不好的地方，欢迎提出共同研究，也欢迎大家有更好的方案可以跟我分享研究！

感谢你的阅读！

原文链接：<https://segmentfault.com/a/1190000003931773>