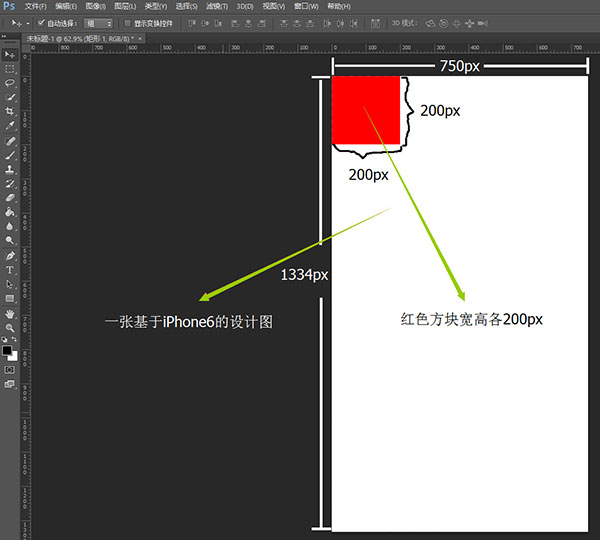
移动端开发的干货篇

之前写了一篇文章[《一篇真正教会你开发移动端一面的文章(一)》](http://hcysun.me/2015/10/16/%E4%B8%80%E7%AF%87%E7%9C%9F%E6%AD%A3%E6%95%99%E4%BC%9A%E4%BD%A0%E5%BC%80%E5%8F%91%E7%A7%BB%E5%8A%A8%E7%AB%AF%E9%A1%B5%E9%9D%A2%E7%9A%84%E6%96%87%E7%AB%A0(%E4%B8%80)/)。那是本篇文章的基础，如果没有阅读过的同学可以去看看，今天就给大家带来干货，真真正正的讲到如何很好的开发一个移动端的页面



好了，让我们开始吧，从哪里开始呢？从设计图开始，即PSD稿件：  
移动端PSD稿件的尺寸肯定较之PC端的PSD稿件不同，具体体现在设计图的尺寸上，现在移动端的设计图尺寸大多以iPhone5和iPhone6的设备像素尺寸作为依据，比如拿到一张PSD设计图，它的总宽度为640px(iPhone5)或者750px(iPhone6)。本例就拿iPhone6的设计图尺寸为标准进行讲解，其它设计图尺寸道理是一样的，这并不影响我们的开发。

首先我们要有一张设计图才行，看下图，假设我们有一张设计图，它很简单，只有一个红色的方块：



拿到了设计图，于是你开开心心的开始写代码了，你打开了编辑器，并写下了如下HTML代码：

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" />

</head>

<body>

<div class="box"></div>

</body>

</html>

HTML代码写好了，你用了一个带有box类的div标签作为ps稿中的红色块，经过尺寸测量，你为上面代码添加了CSS样式，最后你的代码是这样的：

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" />

<style>

body{

margin: 0;

padding: 0;

}

.box{

width: 200px;

height: 200px;

background: red;

}

</style>

</head>

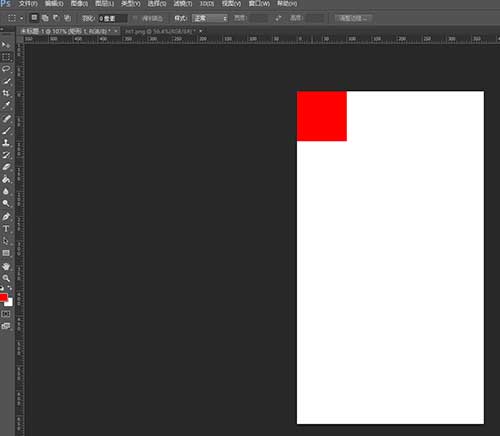
<body>

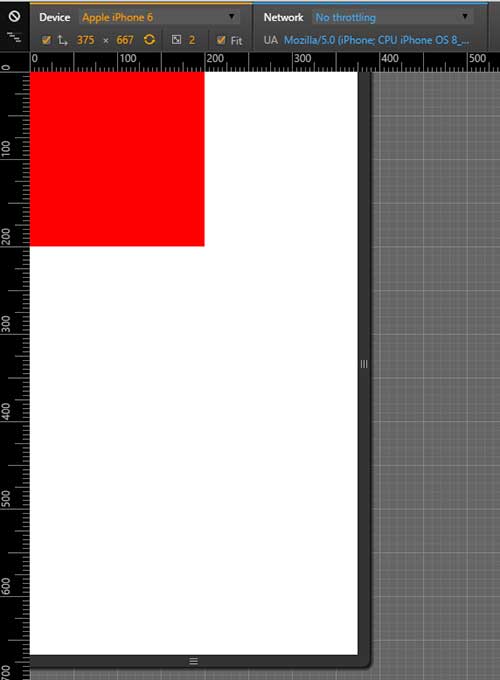
<div class="box"></div>

</body>

</html>

上面的代码中，你只是在原来的基础上增加了CSS样式，首先你清除了body标签上的默认样式，这个没什么好说的，然后你根据设计图中测量的尺寸来给box编写样式，宽200px;高200px;背景红色。看上去并没有什么问题，于是你开开心心的打开浏览器，刷新页面，你的脸色沉了下来，因为你看到了你不想看到的结果，如下图，上图为设计稿的样式，下图为你编写的html文件的样式：

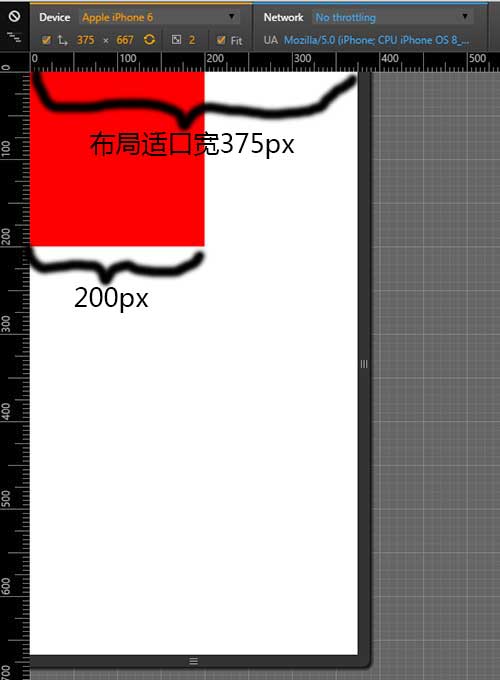




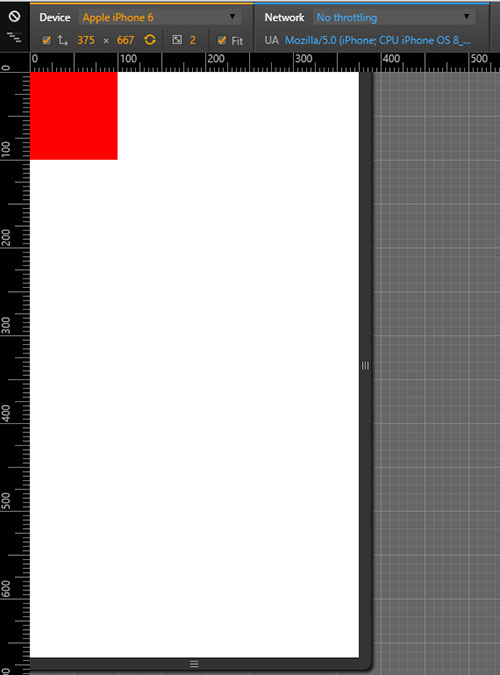
通过对比psd原稿和我们目前所写的html页面，可以看出我们html页面的问题，红色方块与整个页面的比例和psd原稿不一样啊，那么为什么我们明明是按照原稿测量的尺寸写出来的代码却和psd原稿显示的效果不一样呢？别忘了，psd原稿的尺寸是按照设备像素设计的，由于我们所用的设计稿是基于iPhone6设计的，所以我们设计稿的尺寸就是iPhone6的设备像素的尺寸，也就是750px，而我们CSS中的样式是基于**布局视口**的尺寸计算的，由于我们html页面中由于写入了以下meta标签：

<meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" />

在上一篇我们讲过， width=device-width 这段代码是让布局视口的尺寸等于理想视口。  
根据公式(缩放比例为1)：  
**设备像素比（DPR） = 设备像素个数 / 理想视口像素个数（device-width）**  
因为iPhone6的DPR（设备像素比）为2，设备像素为750，所以iPhone6的理想视口尺寸为375px。所以上面代码最终导致的是：使我们布局视口的宽度变成了375px。而我们CSS中编写的样式尺寸又是基于布局视口计算的，所以我们得到的页面看上去比例不对，如下图：


如上面两幅图片，我们知道，psd稿的总宽是750px，元素宽200px，而我们真正做页面的时候，布局视口的宽度是375px，正好是设计稿的一半。所以我们不能直接使用设计稿上面测量所得的像素尺寸，根据比例，我们应该将测量所得的尺寸除以2，才是我们CSS中布局所用的尺寸，据此，我们将200px除以2得到100px，于是我们修改代码，将红色方块的宽高都设为100px，刷新页面，看看比例是不是和设计图一样了？答案是肯定的，如下图为修改后的html页面：



这样，我们就得到了正确的数据，并且正确的写出了页面，你很高兴，可是问题来了，如果你在做页面的时候，测量了一个元素的宽度，宽度是一个奇数，比如111像素，按照我们之前的做法是，将测量到的数据除以2，得到我们真正使用的数据，所以111除以2等于55.5px，我们知道，计算机(手机)没办法显示不到一个像素的像素值，计算机(手机)会自动将其补全为一个像素进行显示，所以最终会将元素显示为56像素，这并不是我们想要的结果。  
另外，我们的设计稿是基于iphone6设计的，我们调试页面也是在iphone6下调试的。又因为iphone6的设备像素比试2，所以我们才能由设计稿测量的数据除以2后直接使用，并且在iphone6下没有问题，但是你要知道，并不是所有手机的设备像素比都是2，有的手机的设备像素比试2.5或者3。并且不同设备的设备像素又不同，这样就导致理想视口的尺寸不同，从而导致布局视口的尺寸不同，那么我们直接根据iphone6的设计稿尺寸除以2得到的尺寸用来编写CSS是不能在所有设备下完好显示的。

所以，我们要换一个方法。  
于是我们想到：如果我们能将布局视口的尺寸设置为和设备像素尺寸相等的话，这样我们就保证了设计图与页面的1:1关系，那么我们就可以直接使用psd中测量的尺寸了，然后在其他尺寸的手机中，我们进行等比缩放就ok了。那么如何才能让布局视口的尺寸等于设备像素尺寸呢？

我们注意到meta标签中的 width=device-width 这段代码，首先你要明白这句话的意思，前面讲过，这句话最终导致的结果是：让布局视口的尺寸等于理想视口的尺寸。言外之意就是,在代码 width=device-width 中：

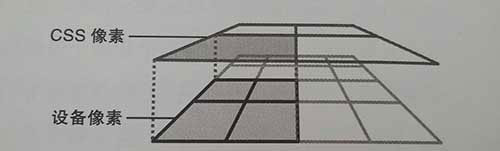
width：是布局视口的width  
device-width：是理想视口的宽度

根据公式(缩放比例为1)：

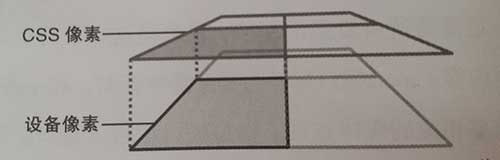
**设备像素比（DPR） = 设备像素个数 / 理想视口像素个数（device-width）**

以iphone6为例：  
设备像素比（DPR）：2  
设备像素个数：750  
所以在缩放比例为1的情况下，iphone6理想视口的像素个数为 750 / 2 = 375，也就是说，对于iphone6来讲 device-width的值为375

所以我们通过width=device-width这句话，间接的将布局视口的尺寸设为了375，也就是说，如果我们能改变理想视口的尺寸，也就改变了布局适口的尺寸，如何改变理想视口的尺寸呢？这就要讲到缩放了，上一篇我们讲到过缩放，缩放是缩小或放大CSS像素的过程，以iphone6为例，当我们缩放比例为1:1的时候，由于iphone6的设备像素比为2，所以iphone6的设备像素与CSS像素的关系看起来就像下图这样：



一个CSS像素宽度等于两个设备像素宽度，所以750px的设备宽度的布局视口为357CSS像素。这是在缩放比例为1的情况下，既然缩放可以放大或缩小CSS像素，所以如果我们将CSS像素的宽度缩放至与设备像素宽度相等了，那么750个设备像素也就能显示750个CSS像素，缩放后的设备像素与CSS像素看起来应该像下图这样：



但是，我们的缩放倍数是多少呢？在缩放比例为1的时候，一个CSS像素的宽度 = 两个设备像素的宽度，如果我们想让 一个CSS像素的宽度 = 一个设备像素的宽度，我们就要将CSS像素缩小为原来的0.5倍，实际上，我们缩小的倍数 = 设备像素比的倒数。  
于是，我们修改上面的HTML代码(修改了meta标签)：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=0.5,maximum-scale=0.5,user-scalable=no" />

<style>

body{

margin: 0;

padding: 0;

}

.box{

width: 200px;

height: 200px;

background: red;

}

</style>

</head>

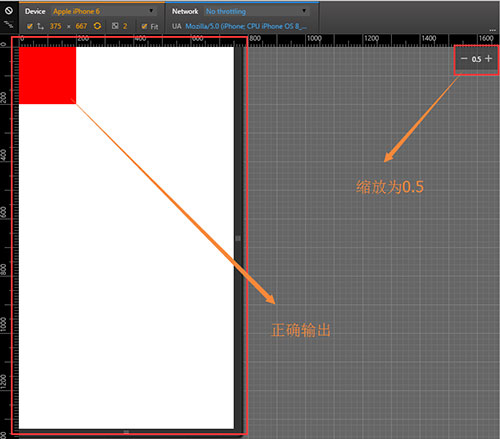
<body>

<div class="box"></div>

</body>

</html>

注意，上面代码中我们给红色方块使用的CSS尺寸直接使用的是psd稿中测量的尺寸，我们刷新页面，怎么样？满意吧：



但是我们这是有个前提的，那就是缩放0.5倍只适用于设备像素比为2的设备（因为缩放值 = 1 / 设备像素比）。所以，为了适应所有的设备，我们应该用javascript代码动态生成meta标签：

var scale = 1 / window.devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','width=device-width,initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

其中 **window.devicePixelRatio** 的值为设备像素比。  
于是我们的代码变成了这样：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="" />

<style>

body{

margin: 0;

padding: 0;

}

.box{

width: 200px;

height: 200px;

background: red;

}

</style>

</head>

<body>

<div class="box"></div>

<script>

var scale = 1 / window.devicePixelRatio;

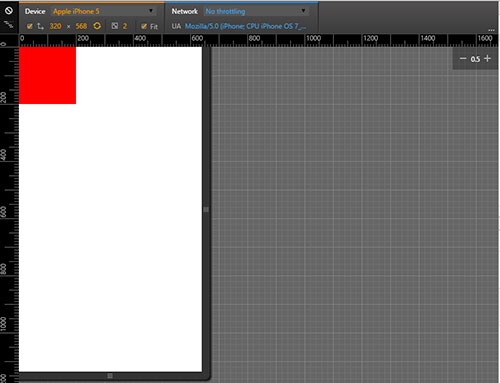
document.querySelector('meta[name="viewport"]').setAttribute('content','width=device-width,initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

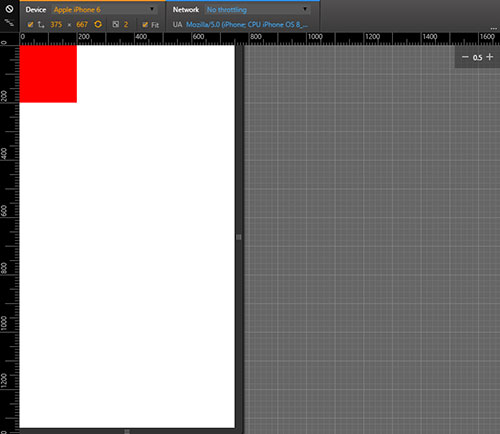
</script>

</body>

</html>

上面的代码最终能保证一个问题，那就是无论任何设备，布局视口的宽度总是等于设备像素。  
这样，我们在设计图中测量为200px的宽度就能直接用在CSS中了，并且在iphone6中显示完好，可是别忘了，我们的设计图就是根据iphone6设计的，如果换做其他设备，还能显示完好么？我们不妨试一下，如下图，是上面代码在iphone5和iphone6下的对比：





我们发现，无论是五还是6，即使设备像素变了，即屏幕宽度变了，可是红色方块的宽度并没有变，这并不是一个好的现象，因为这样页面的元素就不成比例了，会影响到布局，所以我们要想办法让我们页面的元素跟着设备变化而等比缩放，这就是我们要解决的第二个问题，怎么实现呢？这就要讲到rem的知识点了。

rem

什么是rem？  
rem是相对尺寸单位，相对于html标签字体大小的单位，举个例子：  
如果html的font-size = 18px;  
那么1rem = 18px，需要记住的是，rem是基于html标签的字体大小的。

相信你已经明白了，对没错，我们要把之前用px做元素尺寸的单位换成rem，所以，现在的问题就是如果转换，因为rem是根据html标签的font-size值确定的，所以我们只要确定html标签的font-size值就行了，我们首先自己定一个标准，就是让font-size的值等于设备像素的十分之一，即：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 10 + 'px';

以iphone6为例，html标签的font-size的值就等于 750 / 10 = 75px 了，这样 1rem = 75px，所以红色方块200px换算为rem单位就是 200 / 75 = 2.6666667rem。  
那么在iphone5中呢？因为iphone5的设备像素为640，所以iphone的html标签的font-size的值为 640 / 10 = 64px，所以 1rem = 64px，所以在iphone6中显示为200px的元素在iphone5中会显示为 2.6666667 \* 64 像素，这样，在不同设备中就实现了让元素等比缩放从而不影响布局。而上面的方法也是手机淘宝所用的方法。所以，现在你只需要将你测量的尺寸数据除以75就转换成了rem单位，如果是iPhone5就要除以64，即除以你动态设置的font-size的值。

另外需要注意的是：做页面的时候文字字体大小不要用rem换算，还是使用px做单位。后面会讲到。

让我们来总结一下我们现在了解的方法：

1、将布局视口大小设为设备像素尺寸：

var scale = 1 / window.devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','width=device-width,initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

2、动态设置html字体大小：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 10 + 'px';

3、将设计图中的尺寸换算成rem

元素的rem尺寸 = 元素的psd稿测量的像素尺寸 / 动态设置的html标签的font-size值

说了一大堆，其实我们使用下面的html莫板就可以写页面了，唯一需要你做的就是计算元素的rem尺寸，所以即使你没看懂上面的讲述也不重要，你只要将莫板拿过去用就好了：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="" />

</head>

<body>

<script>

var scale = 1 / window.devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','width=device-width,initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

document.documentElement.style.fontSize = document.documentElement.clientWidth / 10 + 'px';

</script>

</body>

</html>

现在我们使用上面的方法修改我们的代码如下：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="" />

<style>

body{

margin: 0;

padding: 0;

}

.box{

width: 2.66666667rem;

height: 2.66666667rem;

background: red;

}

</style>

</head>

<body>

<div class="box"></div>

<script>

var scale = 1 / window.devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','width=device-width,initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

document.documentElement.style.fontSize = document.documentElement.clientWidth / 10 + 'px';

</script>

</body>

</html>

打开浏览器，分别在iPhone6和iPhone5下查看页面，我们会发现，现在的元素可以根据手机的尺寸不同而等比缩放了。

上面的方法是手机淘宝的方法，有一个缺点，就是转化rem单位的时候，需要除以font-size的值，淘宝用的是iPhone6的设计图，所以淘宝转换尺寸的时候要除以75，这个值可不好算，所以还要借用计算器来完成，影响开发效率，另外，在转还rem单位时遇到除不尽的数时我们会采用很长的近似值比如上面的2.6666667rem，这样可能会使页面元素的尺寸有偏差。

除了上面的方法比较通用之外，还有一种方式，我们来重新思考一下：

上面做页面的思路是：拿到设计图，比如iPhone6的设计图，我们就将浏览器设置到iPhone6设备调试，然后使用js动态修改meta标签，使布局视口的尺寸等于设计图尺寸，也就是设备像素尺寸，然后使用rem替代px做尺寸代为，使得页面在不同设备中等比缩放。

现在假如我们不去修改meta标签，正常使用缩放为1:1的meta标签，即使用如下meta标签：

<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no" />

还以iPhone6为例，我们知道，在缩放为1:1的情况下，根据公式：

**设备像素比（DPR） = 设备像素个数 / 理想视口像素个数（device-width）**

我们知道：  
设备像素 = 设计图尺寸 = 750px  
布局视口 = 375px

假设我们以iPhone6设计图尺寸为标准，在设计图的尺寸下设置一个font-size值为100px。  
也就是说：750px宽的页面，我们设置100px的font-size值，那么页面的宽度换算为rem就等于 750 / 100 = 7.5rem。

我们就以页面总宽为7.5rem为标准，那么在布局视口中，也就是页面总宽为375px下，font-size值应该是多少？很简单：

font-size = 375 / 7.5 = 50px

那么在iPhone5下呢？因为iPhone5的布局视口宽为320px，所以如果页面总宽以7.5为标准，那么iPhone5下我们设置的font-size值应该是：

font-size = 320 / 7.5 =42.666666667px

也就是说，不管在什么设备下，我们都可以把页面的总宽度设为一个以rem为单位的定值，比如本例就是7.5rem，只不过，我们需要根据布局视口的尺寸动态设置font-size的值：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 7.5 + 'px';

这样，无论在什么设备下，我们页面的总宽度都是7.5rem，所以我们直接在设计图上测量px单位的尺寸，然后除以100转换成rem单位后直接使用就可以了，比如，在iPhone6设计图中测量一个元素的尺寸为200px，那么转换成rem单位就是 200 / 100 = 2rem，因为在不同设备下我们动态设置了html标签的font-size值，所以不同设备下相同的rem值对应的像素值是不同的，这样就实现了在不同设备下等比缩放。我们修改html代码如下：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no" />

<style>

body{

margin: 0;

padding: 0;

}

.box{

width: 2rem;

height: 2rem;

background: red;

}

</style>

</head>

<body>

<div class="box"></div>

<script>

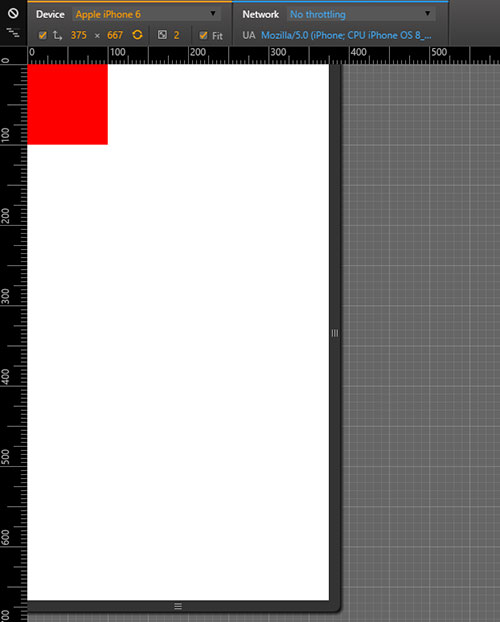
document.documentElement.style.fontSize = document.documentElement.clientWidth / 7.5 + 'px';

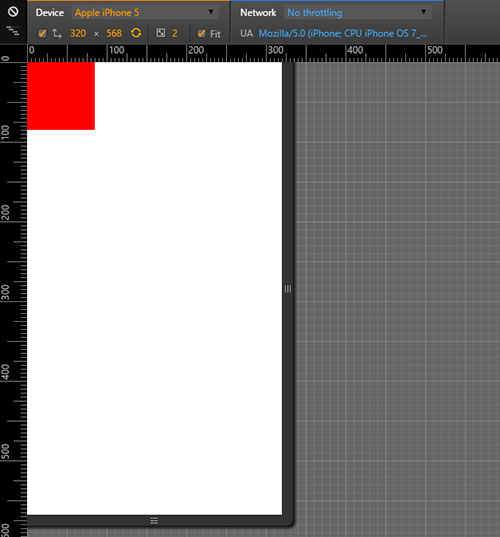
</script>

</body>

</html>

刷新页面，分别在iPhone6和iPhone5下调试查看结果，会发现如下图，使我们想要的效果，等比缩放，ok，实际上这种做法也是网易的做法：





下面，我们来总结一下第二种做法：

1、拿到设计图，计算出页面的总宽，为了好计算，取100px的font-size，如果设计图是iPhone6的那么计算出的就是7.5rem，如果页面是iPhone5的那么计算出的结果就是6.4rem。  
2、动态设置html标签的font-size值：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 以rem为单位的页面总宽 + 'px';

如iPhone6的设计图就是：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 7.5 + 'px';

iPhone5的设计图就是：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 6.4 + 'px';

3、做页面是测量设计图的px尺寸除以100得到rem尺寸。  
4、和淘宝的做法一样，文字字体大小不要使用rem换算。

下面是这种做法的html莫板：

<html>

<head>

<title></title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no" />

</head>

<body>

<script>

document.documentElement.style.fontSize = document.documentElement.clientWidth / 7.5 + 'px';

</script>

</body>

</html>

由于这种做法在开发中换算rem单位的时候只需要将测量的尺寸除以100即可，所以不需要使用计算器我们就可以很快的完成计算转换，所以这也会提升开发效率，本人也比较青睐这种做法。

另外，无论是第一种做法还是第二种做法，我们都提到了，文字字体大小是不要换算成rem做单位的，而是使用媒体查询来进行动态设置，比如下面的代码就是网易的代码：

代码片段一：

@media screen and (max-width: 321px) {

body {

font-size:16px

}

}

@media screen and (min-width: 321px) and (max-width:400px) {

body {

font-size:17px

}

}

@media screen and (min-width: 400px) {

body {

font-size:19px

}

}

代码片段二：

@media screen and (max-width: 321px) {

header,footer {

font-size:16px

}

}

@media screen and (min-width: 321px) and (max-width:400px) {

header,footer {

font-size:17px

}

}

@media screen and (min-width: 400px) {

header,footer {

font-size:19px

}

}

我们总结一下网易在文字字体大小上的做法，在媒体查询阶段，分为三个等级分别是：  
321px以下  
321px - 400px之间  
400px以上

具体文字大小要多少个像素这个以设计图为准，但是这三个等级之间是有规律的，仔细观察发现，321px以下的屏幕字体大小比321px - 400px之间的屏幕字体大小要小一个像素，而321px - 400px之间的屏幕字体大小要比400以上屏幕字体大小要小2个像素。依照这个规律，我们根据设计图所在的像素区段先写好该区段的字体大小，然后分别写出另外两个区段的字体大小媒体查询代码就可以了。

终于码完了这第二篇文章，无力再多说其他的话，望对大家有帮助，有些细节地方没有细说，另外作者水平有限，希望大家指正共同进步，谢谢。