[移动端适配：font-size设置的思考](https://www.cnblogs.com/axl234/p/5156956.html)

**1. 问题的引出**

如果html5要适应各种分辨率的移动设备，可以使用rem这样的尺寸单位，针对各个分辨率范围在html上设置font-size的代码：

[复制代码](javascript:void(0);)

复制代码

html{font-size:10px}

@media screen and (min-width:321px) and (max-width:375px){html{font-size:11px}}

@media screen and (min-width:376px) and (max-width:414px){html{font-size:12px}}

@media screen and (min-width:415px) and (max-width:639px){html{font-size:15px}}

@media screen and (min-width:640px) and (max-width:719px){html{font-size:20px}}

@media screen and (min-width:720px) and (max-width:749px){html{font-size:22.5px}}

@media screen and (min-width:750px) and (max-width:799px){html{font-size:23.5px}}

@media screen and (min-width:800px){html{font-size:25px}}

复制代码

[复制代码](javascript:void(0);)

在实际项目中，把与元素尺寸有关的css，如width,height,line-height,margin,padding等都以rem作为单位，这样页面在不同设备下就能保持一致的网页布局。

举例来说，网页有一个.item类，设置了width为3.4rem，该类在不同分辨率下对应的实际宽度如下：

[复制代码](javascript:void(0);)

复制代码

321px <= device-width <= 375px，font-size:11px ---> .item的width：34px

376px <= device-width <= 414px，font-size:12px ---> .item的width：37.4px

415px <= device-width <= 639px，font-size:15px ---> .item的width：40.8px

640px <= device-width <= 719px，font-size:20px ---> .item的width：51px

720px <= device-width <= 749px，font-size:22.5px ---> .item的width：76.5px

750px <= device-width <= 799px，font-size:23.5px ---> .item的width：79.8999999px

800px <= device-width ，font-size:25px ---> .item的width：85px

复制代码

[复制代码](javascript:void(0);)

以上代码乍看没啥问题，响应式设计不就应该是这么干的吗？但是从工作量和复杂度方面来考虑，它有以下几个不足：

* （1）.item类在所有设备下的width都是3.4rem，但在不同分辨率下的实际像素是不一样的，所以在有些分辨率下，width的界面效果不一定合适，有可能太宽，
* 有可能太窄，这时候就要对width进行调整，那么就需要针对.item写媒介查询的代码，为该分辨率重新设计一个rem值。然而，这里有7种媒介查询的情况，css
* 又有很多跟尺寸相关的属性，哪个属性在哪个分辨率范围不合适都是不定的，最后会导致要写很多的媒介查询才能适配所有设备，而且在写的时候rem都得根据某个
* 分辨率html的font-size去算，这个计算可不见得每次都那么容易，比如40px / 23.5px，这个rem值口算不出来吧！由此可见这其中的麻烦有多少。
* （2）以上代码中给出的7个范围下的font-size不一定是合适的，这7个范围也不一定合适，实际有可能不需要这么多，所以找出这些个范围，以及每个范围最合
* 适的font-size也很麻烦
* （3）设计稿都是以分辨率来标明尺寸的，前端在根据设计稿里各个元素的像素尺寸转换为rem时，该以哪个font-size为准呢？这需要去写才能知道。

正是因为以上提到的一些不足，我觉得这种适配方式不是特别好，写起来太麻烦。为了完成工作，我们需要找寻更简单更有效率的方法。那么html5该如何去做众多移动设备

的适配呢？我目前已知的有3种解决方法，将会在下文的第2,3,4部分阐述，如果你阅读之后，有什么想法，尽可在评论中与我交流。

**2. 简单问题简单解决**

我觉得有些web app并一定很复杂，比如拉勾网，你看看它的页面在iphone4,iphone6,ipad下的样子就知道了：

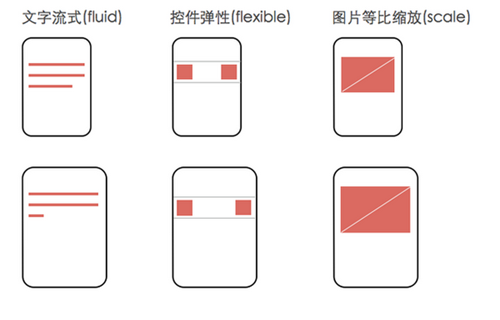
[[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135809163-1797857351.png)](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135809163-1797857351.png)[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135810335-1567896258.png)

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135814632-303159622.png)

它的页面有一个特点，就是：

* 顶部与底部的bar不管分辨率怎么变，它的高度和位置都不变
* 中间每条招聘信息不管分辨率怎么变，招聘公司的图标等信息都位于条目的左边，薪资都位于右边

这种app是一种典型的弹性布局：关键元素高宽和位置都不变，只有容器元素在做伸缩变换。对于这类app，记住一个开发原则就好：文字流式，控件弹性，图片等比缩放。以图描述：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135815944-853042582.png)

这个规则是一套基本的适配规则，对于这种简单app来说已经足够，同时它也是后面要说的rem布局的基础。另外对于拉勾这种app可能需要额外媒介查询对布局进行调整的就是

小屏幕设备。举例来说，因为现在很多设计稿是根据iphone6的尺寸来的，而iphon6设备宽的逻辑的像素是375px，而iphone4的逻辑像素是320个像素，所以如果你根据设

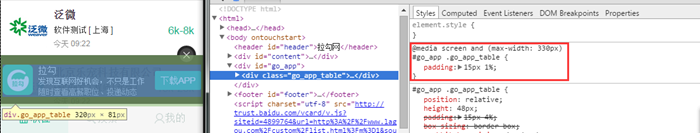
计稿做出来的东西，在iphone4里面可能显示不下，比如说拉钩网底部那个下载框，你对比看下就知道了，这是4：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135817241-908333531.png)

这是6：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135818241-453683920.png)

6下面两边的间距比4多很多，说明拉勾对4肯定是做过适配的，从代码也可以证实这一点：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135819460-508552007.png)

不过如果你拿到的是根据4的设计稿，那就没有问题，比4分辨率大的设备肯定能显示根据4的尺寸做出来的东西。

还有一点，这种情况css尺寸单位用px就好，不要用rem，避免增加复杂度。

**3. 网易的做法**

先来看看网易在不同分辨率下，呈现的效果：

[[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135821601-181079685.png)](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135821601-181079685.png)[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135824788-186132405.png)

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135827335-740417803.png)[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135829991-1902051127.png)

从上面几张图可以看出，随着分辨率的增大，页面的效果会发生明显变化，主要体现在各个元素的宽高与间距。375\*680的比320\*680的导航栏明显要高。能够达到这种效果的根本原因就是因为网易页面里

除了font-size之外的其它css尺寸都使用了rem作为单位，比如你看导航栏的高度设置代码：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135833163-1178991927.png)

可是在本文第1部分提到，使用rem布局结合在html上根据不同分辨率设置不同font-size有很多不好解决的麻烦，网易是如何解决的呢？**最根本的原因在于，网易页面上html的font-size不是预先通过媒**

**介查询在css里定义好的，而是通过js计算出来的**，所以当分辨率发生变化时，html的font-size就会变，不过这得在你调整分辨率后，刷新页面才能看得到效果。你看代码就知道为啥font-size是直接写到html

的style上面的了（js设置的原因）：

[image](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135833991-1139531573.png)

它是根据什么计算的，这就跟设计稿有关了，拿网易来说，它的设计稿应该是基于iphone4或者iphone5来的，所以它的设计稿竖直放时的横向分辨率为640px，为了计算方便，取一个100px的font-size为参照，那

么body元素的宽度就可以设置为width: 6.4rem，于是html的font-size=deviceWidth / 6.4。这个deviceWidth就是viewport设置中的那个deviceWidth。根据这个计算规则，可得出本部分开始的四张截图中

html的font-size大小如下：

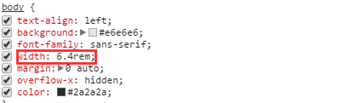
deviceWidth = 320，font-size = 320 / 6.4 = 50px

deviceWidth = 375，font-size = 375 / 6.4 = 58.59375px

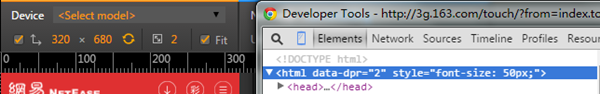
deviceWidth = 414，font-size = 414 / 6.4 = 64.6875px

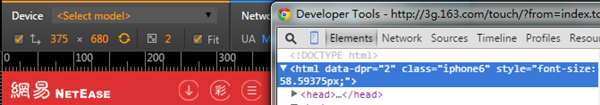
deviceWidth = 500，font-size = 500 / 6.4 = 78.125px

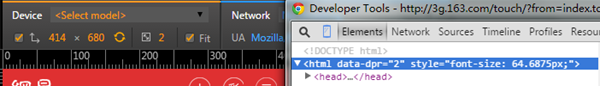
事实上网易就是这么干的，你看它的代码就知道，body元素的宽是：

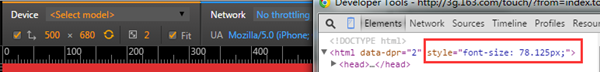
[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135834819-1543091953.png)

根据这个可以肯定它的设计稿竖着时的横向分辨率为640。然后你再看看网易在分辨率为320\*680，375\*680，414\*680，500\*680时，html的font-size是不是与上面计算的一致：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135836210-119149983.png)320\*680

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135837319-1087313867.png)375\*680

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135838476-873400029.png)414\*680

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135839772-1714569449.png)500\*680

这个deviceWidth通过document.documentElement.clientWidth就能取到了，所以当页面的dom ready后，做的第一件事情就是：

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

这个6.4怎么来的，当然是根据设计稿的横向分辨率/100得来的。下面总结下网易的这种做法：

* （1）先拿设计稿竖着的横向分辨率除以100得到body元素的宽度：
* 如果设计稿基于iphone6，横向分辨率为750，body的width为750 / 100 = 7.5rem

如果设计稿基于iphone4/5，横向分辨率为640，body的width为640 / 100 = 6.4rem

* （2）布局时，设计图标注的尺寸除以100得到css中的尺寸，比如下图：
* [](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135841179-598716573.png)
* 播放器高度为210px，写样式的时候css应该这么写：height: 2.1rem。之所以取一个100作为参照，就是为了这里计算rem的方便！
* （3）在dom ready以后，通过以下代码设置html的font-size:

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

* 6.4只是举个例子，如果是750的设计稿，应该除以7.5。
* （4）font-size可能需要额外的媒介查询，并且font-size不能使用rem，如网易的设置：

[复制代码](javascript:void(0);)

复制代码

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

.m-navlist{font-size:15px}

}

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

.m-navlist{font-size:16px}

}

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

.m-navlist{font-size:18px}

}

复制代码

[复制代码](javascript:void(0);)

最后还有2个情况要说明：

第一，如果采用网易这种做法，视口要如下设置：

<meta name="viewport" content="initial-scale=1,maximum-scale=1, minimum-scale=1">

第二，当deviceWidth大于设计稿的横向分辨率时，html的font-size始终等于横向分辨率/body元素宽：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135842976-163350007.png)640\*680

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135844741-479724363.png)641\*680

之所以这么干，是因为当deviceWidth大于640时，则物理分辨率大于1280（这就看设备的devicePixelRatio这个值了），应该去访问pc网站了。事实就是这样，你从手机访问网易，

看到的是触屏版的页面，如果从pad访问，看到的就是电脑版的页面。如果你也想这么干，只要把总结中第三步的代码稍微改一下就行了：

var deviceWidth = document.documentElement.clientWidth;

if(deviceWidth > 640) deviceWidth = 640;

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

**4. 淘宝的做法**

看看淘宝在不同分辨率下，呈现的效果：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135845929-2021863186.png)[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135847804-973770891.png)[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135849491-1942258919.png)

淘宝的效果跟网易的效果其实是类似的，随着分辨率的变化，页面元素的尺寸和间距都相应变化，这是因为淘宝的尺寸也是使用了rem的原因。在介绍它的做法之前，先来了解一点

关于viewport的知识，通常我们采用如下代码设置viewport:

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

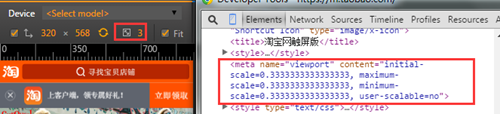
这样整个网页在设备内显示时的页面宽度就会等于设备逻辑像素大小，也就是device-width。这个device-width的计算公式为：

设备的物理分辨率/(devicePixelRatio \* scale)，在scale为1的情况下，device-width = 设备的物理分辨率/devicePixelRatio 。

devicePixelRatio称为设备像素比，每款设备的devicePixelRatio都是已知，并且不变的，目前高清屏，普遍都是2，不过还有更高的，比如2.5, 3 等，我魅族note的手机的devicePixelRatio就是3。

淘宝触屏版布局的前提就是viewport的scale根据devicePixelRatio动态设置：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135852413-46323915.png)在devicePixelRatio为2的时候，scale为0.5

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135853757-310485369.png)在devicePixelRatio为3的时候，scale为0.3333

这么做目的当然是为了保证页面的大小与设计稿保持一致了，比如设计稿如果是750的横向分辨率，那么实际页面的device-width，以iphone6来说，也等于750，这样的话设计稿上标注的尺寸只要除以

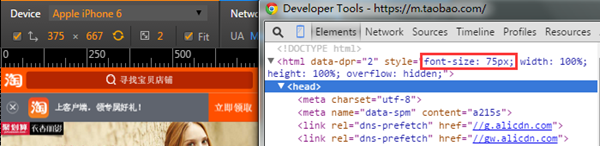
某一个值就能够转换为rem了。通过js设置viewport的方法如下：

var scale = 1 / devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=' + scale + ', maximum-scale=' + scale + ',

minimum-scale=' + scale + ', user-scalable=no');

淘宝布局的第二个要点，就是html元素的font-size的计算公式，font-size = deviceWidth / 10：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135855054-1534792809.png)

接下来要解决的问题是，元素的尺寸该如何计算，比如说设计稿上某一个元素的宽为150px，换算成rem应该怎么算呢？这个值等于设计稿标注尺寸/该设计稿对应的html的font-size。拿淘宝来说的，他们用的设计

稿是750的，所以html的font-size就是75，如果某个元素时150px的宽，换算成rem就是150 / 75 = 2rem。总结下淘宝的这些做法：

* （1）动态设置viewport的scale
* var scale = 1 / devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=' + scale + ', maximum-scale=' + scale + ',

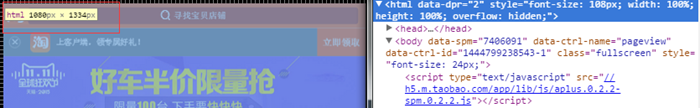
minimum-scale=' + scale + ', user-scalable=no');

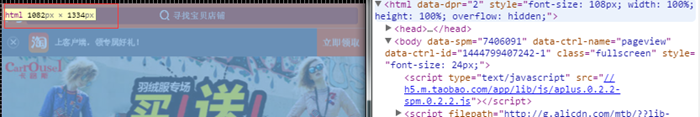
* （2）动态计算html的font-size

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

* （3）布局的时候，各元素的css尺寸=设计稿标注尺寸/设计稿横向分辨率/10
* （4）font-size可能需要额外的媒介查询，并且font-size不使用rem，这一点跟网易是一样的。

最后还有一个情况要说明，跟网易一样，淘宝也设置了一个临界点，当设备竖着时横向物理分辨率大于1080时，html的font-size就不会变化了，原因也是一样的，分辨率已经可以去访问电脑版页面了。

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135857397-1129920815.png)

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135859210-1915975012.png)

**关于这种做法的具体实现，淘宝已经给我们提供了一个开源的解决方案，具体请查看：**

[**https://github.com/amfe/lib-flexible**](https://github.com/amfe/lib-flexible)

之前没有找到这相关的资料，实在不好意思：（

**5. 比较网易与淘宝的做法**

共同点：

* 都能适配所有的手机设备，对于pad，网易与淘宝都会跳转到pc页面，不再使用触屏版的页面
* 都需要动态设置html的font-size
* 布局时各元素的尺寸值都是根据设计稿标注的尺寸计算出来，由于html的font-size是动态调整的，所以能够做到不同分辨率下页面布局呈现等比变化
* 容器元素的font-size都不用rem，需要额外地对font-size做媒介查询
* 都能应用于尺寸不同的设计稿，只要按以上总结的方法去用就可以了

不同点

* + 淘宝的设计稿是基于750的横向分辨率，网易的设计稿是基于640的横向分辨率，还要强调的是，虽然设计稿不同，但是最终的结果是一致的，设计稿的尺寸一个公司设计人员的工作标准，每个公司不一样而已
  + 淘宝还需要动态设置viewport的scale，网易不用
  + **最重要的区别就是：网易的做法，rem值很好计算，淘宝的做法肯定得用计算器才能用好了** 。不过要是你使用了less和sass这样的css处理器，就好办多了，以淘宝跟less举例，我们可以这样编写less：

[复制代码](javascript:void(0);)

复制代码

//定义一个变量和一个mixin

@baseFontSize: 75;//基于视觉稿横屏尺寸/100得出的基准font-size

.px2rem(@name, @px){

@{name}: @px / @baseFontSize \* 1rem;

}

//使用示例：

.container {

.px2rem(height, 240);

}

//less翻译结果：

.container {

height: 3.2rem;

}

复制代码

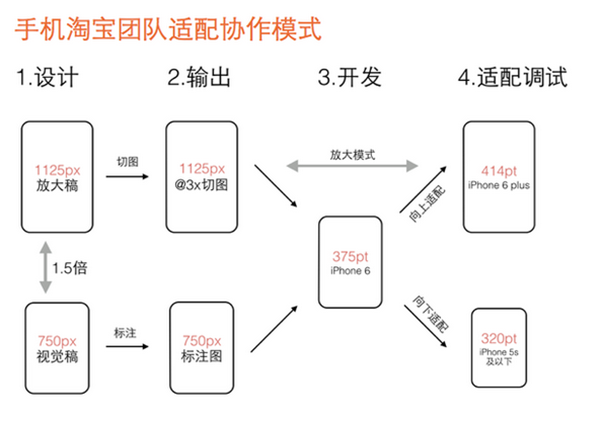
[复制代码](javascript:void(0);)

**6. 如何与设计协作**

前端与设计师的协作应该是比较简单的，最重要的是要规范设计提供给你的产物，通常对于前端来说，我们需要设计师提供标注尺寸后的设计稿以及各种元素的切图文件，有了这些就可以开始布局了。

考虑到Retina显示屏以及这么多移动设备分辨率却不一样的问题，那么设计师应该提供多套设计稿吗？从网易和淘宝的做法来看，应该是不用了，我们可以按照设计稿，先做出一套布局，按照以上方法做适配，

由于是等比适配，所以各个设备的视觉效果差异应该会很小，当然也排除不了一些需要媒介查询特殊处理的情况，这肯定避免不了的。下面这张图是淘宝设计师分享的他们的工作流程：

[](http://images2015.cnblogs.com/blog/459873/201510/459873-20151014135900538-256091655.png)

解释一下就是：

第一步，视觉设计阶段，设计师按宽度750px（iPhone 6）做设计稿，除图片外所有设计元素用矢量路径来做。设计定稿后在750px的设计稿上做标注，输出标注图。

同时等比放大1.5倍生成宽度1125px的设计稿，在1125px的稿子里切图。

第二步，输出两个交付物给开发工程师：一个是程序用到的@3x切图资源，另一个是宽度750px的设计标注图。

第三步，开发工程师拿到750px标注图和@3x切图资源，完成iPhone 6（375pt）的界面开发。此阶段不能用固定宽度的方式开发界面，得用自动布局（auto layout），方便后续适配到其它尺寸。

第四步，适配调试阶段，基于iPhone 6的界面效果，分别向上向下调试iPhone 6 plus（414pt）和iPhone 5S及以下（320pt）的界面效果。由此完成大中小三屏适配。

**注意第三步，就要使用我们以上介绍的网易跟淘宝的适配方法了。**假如公司设计稿不是基于750的怎么办，其实很简单，按上图做一些相应替换即可，但是流程和方法还是一样的。解释一下为什么要在@3x的图里切，

这是因为现在市面上也有不少像魅蓝note这种超高清屏幕，devicePixelRatio已经达到3了，这个切图保证在所有设备都清晰显示。