## **移动端适配（二）实操篇，前端必掌握技能**



程序思维

发布时间：18-04-0500:50

移动端适配（二）实操篇，前端必掌握技

在进行今天的移动端适配实操篇时，先谈及几个知识点：目前主流设计稿为iphone6或iphone8尺寸：750\*1334。设备像素比dpr=window. devicePixelRatio**目前主流的适配移动端页面方法：**用rem做单位，通过css3媒体查询来改变html中的font-size值。rem 是相对于 html 元素的 font-size 的一个单位。如果 html 上定义了 font-size: 20px;，则无论在任何地方都是 1rem = 20px 这个大小不会受到父元素的影响。**实操步骤：**1.加重置样式reset.css/normalize.css确保字体设置及基础样式在各终端有相同的值。2.通过rem的方式适配移动设备，有三种思路：**第一种是：**1) viewport这个meta标签固定如下:<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">2) 通过媒体查询根据设备屏幕尺寸改变html中的font-size值来达到适配移动设备屏幕。典型移动端框架:淘宝的SUI Mobile框架http://m.sui.taobao.org/这个框架统一使用rem对页面进行整体缩放。建议使用者对需要适应页面大小的任何元素都使用 rem 为单位来定义。同时规定iphone6 上使用 1rem = 20px 来换算。小于 375px 的设备上不做缩小处理，对大于375px 宽度的设备进行等比缩放。代码如下：html {font-size: 20px;}@media only screen and (min-width: 400px) {html {font-size: 21.33333333px !important; }}@media only screen and (min-width: 414px) { html {font-size: 22.08px !important; }}@media only screen and (min-width: 480px) { html { font-size: 25.6px !important; }}总结：用SUI Mobile框架的话，UI设计稿应该是iphone6尺寸750\*1334。html的font-size:20px; viewport的meta标签如下：<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0,user-scalable=no">典型移动端网站：京东https://m.jd.com/



代码如下：<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0,user-scalable=no">@media only screen and (min-width:320PX) and (max-width:360PX) {html {font-size: 13.65px}}@media only screen and (min-width:360PX) and (max-width:375PX) {html {font-size: 15.36px}}@media only screen and (min-width:375PX) and (max-width:390PX) {html {font-size: 16px}}@media only screen and (min-width:390PX) and (max-width:414PX) {html {font-size: 16.64px}}@media only screen and (min-width:414PX) and (max-width:640PX) {html {font-size: 17.664px}}@media screen and (min-width:640PX) {html {font-size: 27.31px}}总结：京东UI设计稿是iphone6 的尺寸750\*1334，html的font-size:16px; 设备像素比dpr是2;切页面同学的思维是：UI设计稿上的按钮，量出它的长度，然后该长度除以dpr的值2，得到px的值，然后再除以16算出最终rem的值。**第二种是：**1) viewport这个meta标签根据dpr值变化:<meta name="viewport" content="initial-scale=X, maximum-scale=X, minimum-scale=X, user-scalable=no">至于X是什么,这个由dpr决定。2) 根据最终UI设计稿尺寸，设置html中的font-size字体大小，且字体大小固定不变。然后通过上面的viewport的scale缩放比来达到适配移动设备屏幕。这种思路多需要JS的配合，根据dpr的值不断修改viewport中的scale值。典型移动端网站：美团http://i.meituan.com/



代码如下：<html style=”font-size:100px;” ><meta name="viewport" content="initial-scale=0.5, maximum-scale=0.5, minimum-scale=0.5, width=device-width, user-scalable=no"><script type="text/javascript">//根据屏幕大小及dpi调整缩放和大小 (function() { var scale = 1.0; var ratio = 1; if (window.devicePixelRatio >= 2) { scale \*= 0.5; ratio \*= 2; } var text = '<meta name="viewport" content="initial-scale=' + scale + ', maximum-scale=' + scale +', minimum-scale=' + scale + ', width=device-width, user-scalable=no" />'; document.write(text); document.documentElement.style.fontSize = 50\*ratio + "px"; })();</script>总结：美团UI设计稿是iphone6 的尺寸750\*1334，html的font-size:100px; 设备像素比dpr是2;切页面同学的思维是：UI设计稿上的按钮，量出它的长度，然后该长度除以100算出最终rem的值。然后通过viewport这个meta标签中的scale即1/dpr的值，进行缩小，得以最终适配iphone8手机。**第三种是：**第三种思路和第二种思路相似，唯一不同的是，字体大小也是不固定的。viewport中的缩放比还是由1/dpr决定，字体大小根据媒体查询也在变。这种方式比较少见。典型网站：淘宝https://h5.m.taobao.com



代码如下：<html data-dpr="2" style="font-size: 75px; width: 100%; height: 100%; overflow: hidden;"><meta name="viewport" content="initial-scale=0.5, maximum-scale=0.5, minimum-scale=0.5, user-scalable=no">至于js如何修改viewport中scale的值，由于淘宝代码比较多，就不列举了，大体意思就是：var scale = 1 / devicePixelRatio;document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');总结：淘宝UI设计稿是iphone6 的尺寸750\*1334，html的font-size:75px; 设备像素比dpr是2;切页面同学的思维是：UI设计稿上的按钮，量出它的长度，然后该长度除以75算出最终rem的值。然后通过viewport这个meta标签中的scale即1/dpr的值，进行缩小，得以最终适配iphone8手机。现在移动设备发展很快，我们知道dpr值越大，屏幕越高清。dpr=1的屏基本快没了，iphone4开始已经是retina屏了，除了dpr=2我们常见的设备像素比，其实还有很多android设备的dpr为2.5，还有iphoneX的dpr为3.0。但是我们的设计稿很多时候只有一个尺寸，其实布局还好，就是图片的显示有些问题。关于这个问题，我在移动端适配（一）中已讲到：在retina屏上用1倍图会模糊。普通屏幕下即dpr=1，用两倍图片，肉眼看上去虽然图片不会模糊，但是会觉得图片缺少一些锐利度，或者是有点色差，但可以接受。同样用在超retina屏,dpr=3比如iphone6 plus、iphonex上，就是有点失色，但也可以接受，所以我们只要一个@2x的图即可。如果你想完美，比如dpr=3的设备，图片也是3倍的图片的话，你需要UI设计师配合你，让他出两个尺寸的设计稿，然后前端的人用媒体查询根据不同的dpr,调不同尺寸的图片。所以优秀的UI设计师，多会做矢量图，这样一份设计图就搞定。因为矢量图不会失真，它和分辨率无关。作为前端的你，如果你自己想开发一些页面，没有设计师的话，可以从一些第三方上下载矢量图，目前流行的矢量图有iconfont、svg。关于这些知识点，程序思维在之前的文章中也普及过这些知识，比如：Svg那些事(一)：Svg格式及主流用法和svg那些事(二)：svg sprite和css sprite 、svg和iconfont对比。移动端适配除了上面说的基本方法，还有一些公用的布局方式来配合页面适配。主流布局方法：1. 弹性布局flex2. 媒体查询+流式布局也叫响应式布局3. 百分比布局（巧用）用rem做适配,font-size的值决定我们切图计算是否方便。当然如果你用less,sass这样的东东忽略这个问题。所以就会有 html{font-size:62.5%;} 这样的设置。为什么有这样的百分比？一般浏览器的默认字体是16px（可以通过重置样式，统一设置为16px），所以16\*62.5%=10px，这样1rem=10px，方便切图计算。但这个有兼容性问题，谷歌浏览器能识别的最小字体为12px, 无法再小了,可以在CSS里这样写html{ -webkit-text-size-adjust:none; }解决。有人说谷歌还是有问题，程序思维还没验证，看文章的朋友可以验证下，验证完告诉我们，一块进步。如果这样的话，有人是这样解决的，改为 html{font-size:625%}, 就是 font-size: 100px，这样也很好换算1rem=100px。这样就没有任何兼容性问题了。我平时用的就是SUI框架中的思路，1rem=20px，然后媒体查询修改font-size值，感觉使用还挺方便。你可以选择程序思维总结的目前主流的3种思路中的任何一种。好了，移动端适配的实操篇就说到这，程序思维在平时开发时，也采用上面说的方法，其实这3种思路，都是一种思维，学习就是学习思路。程序思维坚持原创，如要转载一定说明出处，否则后果自负。保护原创，人人有责。喜欢程序思维的朋友，请继续收藏关注我们。