初始移动端

在进行移动端页面重构时,会对页面的尺寸问题模糊不清,移动设备的尺寸非常多,通过这个页面 <http://screensiz.es/phone> 你可以详细的了解目前各个设备尺寸及其他相关的信息.

刚开始接触移动页面重构时,为什么要按照640的大小来设计页面?并且好像只需要制作一套页面,不同的设备就都能适配? 这是为什么呢?不知道耶? 等等… ----这源与对viewport和像素不理解

理解viewport和像素

按如下分类给大家介绍下:

1).什么是像素?

2).什么是屏幕尺寸,屏幕分辨率,屏幕像素密度?

3).什么是ppi,dpi,dp,dip,px?他们之间是什么关系?

4).什么是mdpi,hdpi,xdpi,xxdp?如果计算和区分?

1.什么是像素?

像素是网页布局的基础，但是我们一直在用直觉使用它。

其实存在如下像素：

1. 物理像素(physical pixel)

屏幕的物理像素，一个物理像素是显示器(手机屏幕)上最小的物理显示单元,任何设备屏幕的物理像素的数量都是固定不变

2.设备独立像素(density-independent pixel)单位:dp/dip

设备独立像素(也叫密度无关像素), 可以认为是计算机坐标系统中得一个点，这个点代表一个可以由程序使用的虚拟像素(比如: css像素)，然后由相关系统转换为物理像素,所以物理像素和设备独立像素之间存在着一定的对应关系,这就是接下来讲的设备像素比.

设备像素比 (device pixel ratio)

设备像素比(简称dpr)定义了物理像素和设备独立像素的对应关系，它的值可以按如下的公式的得到：

设备像素比 = 物理像素 / 设备独立像素 // 在某一方向上，x方向或者y方向

javascript中，可以通过window.devicePixelRatio获取到当前设备的dpr。

在css中，可以通过-webkit-device-pixel-ratio，-webkit-min-device-pixel-ratio和 -webkit-max-device-pixel-ratio进行媒体查询，对不同dpr的设备，做一些样式适配

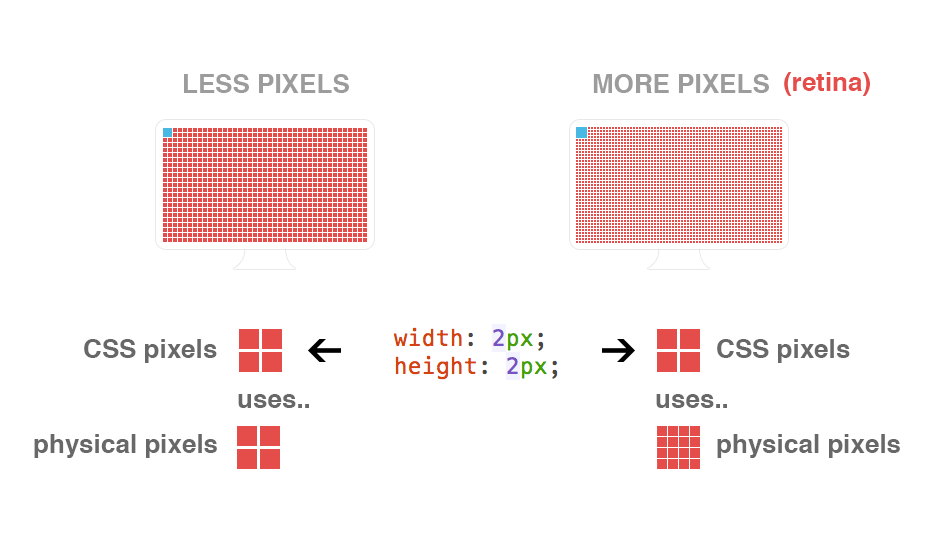
举个例子说下这两个概念:

以iphone6为例：

1. 设备宽高为375×667，可以理解为设备独立像素(或css像素)。

2. dpr为2，根据上面的计算公式，物理像素就应该×2，为750×1334。

用一张图来表现，就是这样(原谅我的盗图)：



上图中可以看出，对于这样的css样式：

width: 2px;

height: 2px;

在不同的屏幕上(普通屏幕 vs retina屏幕)，css像素所呈现的大小(物理尺寸)是一致的，不同的是1个css像素所对应的物理像素个数是不一致的。

在普通屏幕下，1个css像素 对应 1个物理像素(1:1)。 在retina 屏幕下，1个css像素对应 4个物理像素(1:4)。

3.位图像素

一个位图像素是栅格图像(如：png, jpg, gif等)最小的数据单元。每一个位图像素都包含着一些自身的显示信息(如：显示位置，颜色值，透明度等)。

(后面会详细说下retina下图片的展示情况)

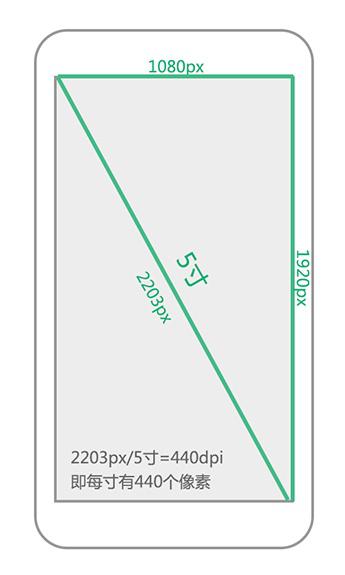
2. 什么是屏幕尺寸、屏幕分辨率、屏幕像素密度？

**屏幕尺寸：**指屏幕的对角线的长度，单位是英寸，1英寸=2.54厘米。常见的屏幕尺寸有2.4、2.8、3.5、3.7、4.2、5.0、5.5、6.0等。

**屏幕分辨率：**指在横纵向上的像素点数，单位是px，1px=1个像素点。一般以纵向像素\*横向像素来表示一个手机的分辨率，如1960\*1080。（这里的1像素值得是物理设备的1个像素点）

**屏幕像素密度：**屏幕上每英寸可以显示的像素点的数量，单位是ppi，即“pixels per inch”的缩写。屏幕像素密度与屏幕尺寸和屏幕分辨率有关，在单一变化条件下，屏幕尺寸越小、分辨率越高，像素密度越大，反之越小。

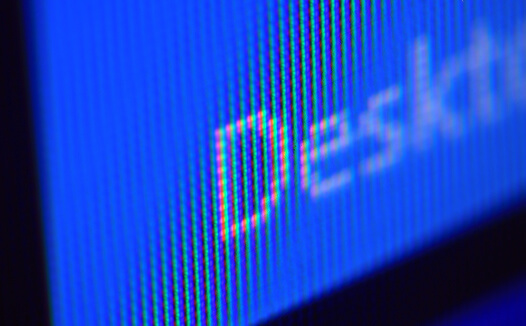
计算像素密度的公式：



勾股定理算出对角线的分辨率：√(1920²+1080²)≈2203px

对角线分辨率除以屏幕尺寸：2203/5≈440dpi。

下面详细的了解下像素密度:

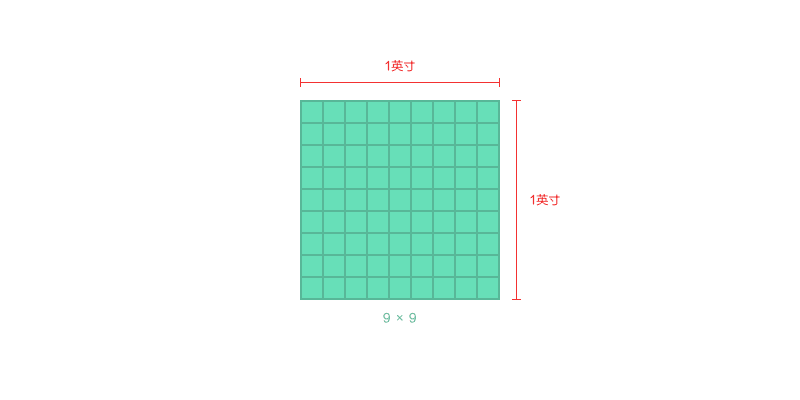


 屏幕是由很多像素点组成的。比如1366x768的屏幕，就是由768行、1366列的像素点组成的。而手机屏幕的物理尺寸，和像素尺寸是不成比例的。最明显的就是iPhone 3GS和iPhone 4的区别。屏幕尺寸一样，屏幕分辨率相差一倍，屏幕像素密度也相差一倍。



所以，我们要引入最重要的一个概念：像素密度，也就是PPI（pixels per inch）

Pixels per inch，准确的说是每英寸的长度上排列的像素点数量。1英寸是一个固定长度，等于2.54厘米，大约是食指最末端那根指节的长度。像素密度越高，代表屏幕显示效果越精细。Retina屏比普通屏清晰很多，就是因为它的像素密度翻了一倍。



4. 什么是mdpi,hdpi,xdpi,xxdp?

Google官方指定按照下列标准区分不同设备的dpi：



如图所示，像素密度在120左右的屏幕归为ldpi，160左右的归为mdpi，以此类推。这样，所有的Android屏幕都找到了自己的位置，并赋予了相应的倍率：

• ldpi [0.75倍]

• mdpi [1倍]

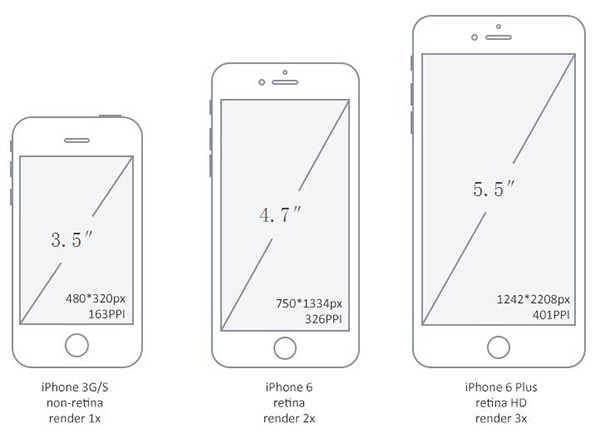
• hdpi [1.5倍]

• xhdpi [2倍]

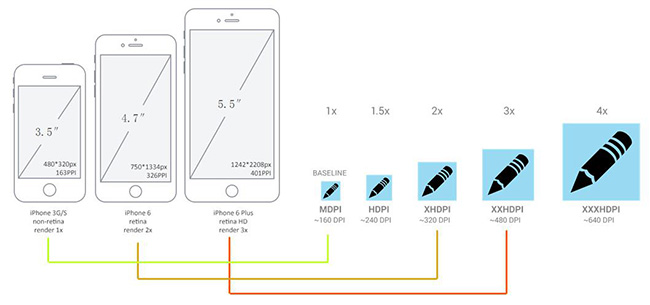
• xxhdpi [3倍]

• xxxhdpi [4倍]

苹果的区分则更为简单：非高清屏、高清屏、超高清屏。



从数值上看，苹果和安卓有这样的对应关系：



单位之间的换算关系随倍率变化：

• 1倍：1pt=1dp=1px（mdpi、iPhone 3gs）

• 1.5倍：1pt=1dp=1.5px（hdpi）

• 2倍：1pt=1dp=2px（xhdpi、iPhone 4s/5/6）

• 3倍：1pt=1dp=3px（xxhdpi、iPhone 6 plus）

• 4倍：1pt=1dp=4px（xxxhdpi）

重构移动页面的时候使用px其实跟安卓开发中使用dp是一样的，有个背后的系数会帮我们把数值适配到这款手机的大小。而这个系数就是上图的1x、2x、3x……这个系数怎么来？是依靠分辨率和屏幕尺寸计算到的屏幕像素密度，再看屏幕像素密度属于哪个系数等级。



单位决定了我们的思考方式。在设计和开发过程中，应该尽量使用逻辑像素尺寸来思考界面。设计Android应用时，有的设计师喜欢把画布设为1080x1920，有的喜欢设成720x1280。给出的界面元素尺寸就不统一了。Android的最小点击区域尺寸是48x48dp，这就意味着在xhdpi的设备上，按钮尺寸至少是96x96px。而在xxhdpi设备上，则是144x144px。

无论画布设成多大，我们设计的是基准倍率的界面样式，而且开发人员需要的单位都是逻辑像素。所以为了保证准确高效的沟通，双方都需要以逻辑像素尺寸来描述和理解界面，无论是在标注图还是在日常沟通中。不要再说“底部标签栏的高度是96像素，我是按照xhdpi做的”这样的话了。

单位决定了我们的思考方式。在设计和开发过程中，应该尽量使用逻辑像素尺寸来思考界面。设计Android应用时，有的设计师喜欢把画布设为1080x1920，有的喜欢设成720x1280。给出的界面元素尺寸就不统一了。Android的最小点击区域尺寸是48x48dp，这就意味着在xhdpi的设备上，按钮尺寸至少是96x96px。而在xxhdpi设备上，则是144x144px。

无论画布设成多大，我们设计的是基准倍率的界面样式，而且开发人员需要的单位都是逻辑像素。所以为了保证准确高效的沟通，双方都需要以逻辑像素尺寸来描述和理解界面，无论是在标注图还是在日常沟通中。不要再说“底部标签栏的高度是96像素，我是按照xhdpi做的”这样的话了

web怎么办?

移动端页面的绝对单位仍然是px，至少代码里这么写，但它的道理也和app一样。由于像素密度是设备本身的固有属性，它会影响到设备中的所有应用，包括浏览器。前端技术可以善加利用设备的像素密度，只需一行代码，浏览器便会使用app的显示方式来渲染页面。根据像素密度，按相应倍率缩放。

以iPhone 5s为例，屏幕的分辨率是640x1136，倍率是2。浏览器会认为屏幕的分辨率是320x568，仍然是基准倍率的尺寸。所以在制作页面时，只需要按照基准倍率来就行了。无论什么样的屏幕，倍率是多少，都按逻辑像素尺寸(实际像素除以倍率，就得到逻辑像素尺寸。只要两个屏幕逻辑像素相同，它们的显示效果就是相同的)来设计和开发页面。只不过在准备资源图的时候，需要准备2倍大小的图，通过代码把它缩成1倍大小显示，才能保证清晰.

Web

手机端网页就没有统一标准了，比较流行的做法是按照iPhone 5的尺寸来设计。倍率2，逻辑像素320x568。

这样的做法比较实在，倍率2的屏幕无论在iOS还是Android方面都是主流，而且又是2倍屏幕中逻辑像素最小的。所以图片的尺寸可以保持在较小的水平，页面加载速度快。当然，缺点就是在倍率3的设备上看，图片不是特别清晰。

如果追求图片质量，愿意牺牲加载速度，那么可以按照最大的屏幕来设计。也就是iPhone 6 plus的尺寸，倍率3，逻辑像素414x736。

总结

移动端的尺寸比PC端复杂，关键就在倍率。但也正因为倍率的存在，把大大小小的屏幕拉回到同一水平线，得以保证一套设计适应各种屏幕。站在这条水平线的角度看，会发现它很好理解。

**Viewport**

视口

桌面浏览器中，浏览器窗口就是约束你的CSS布局视口（又称初始包含块）。它是所有CSS百分比宽度推算的根源，它的作用是给CSS布局限制了一个最大宽度，视口的宽度和浏览器窗口宽度一致。

但是在移动端，情况就很复杂了。

布局视口

一个没有为移动端做优化的网页，会尽可能缩小网页让用户看到所有东西。

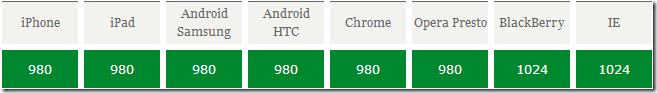
手机浏览器是把页面放在一个虚拟的“窗口”（viewport）中，窗口可大于或小于手机的可视区域，一般手机默认viewport大于可视区域。这样不会破坏没有针对手机浏览器优化的网页的布局，用户可以通过平移和缩放来看网页的其他部分。

浏览器厂商为了让用户在小屏幕下网页也能够显示地很好，所以把视口宽度设置地很大，一般在 768px ~ 1024px 之间，最常见的宽度是 980px。

所以，在手机上，视口与移动端浏览器屏幕宽度不再相关联，是完全独立的，这个浏览器厂商定的视口被称为布局视口。



部分机型默认viewport大小：



布局视口我们是看不见的，最常见的宽度是980 ，并且被缩放在了屏幕内。

我们可能会这样设置viewport：

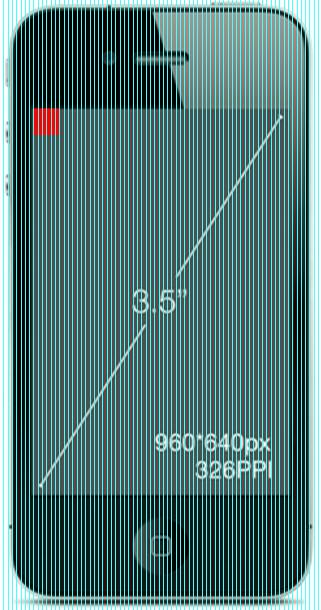
http://ossweb-img.qq.com/upload/webplat/info/tgideas/201509/1441639177_1436653066_26259_imageAddr.jpg

****

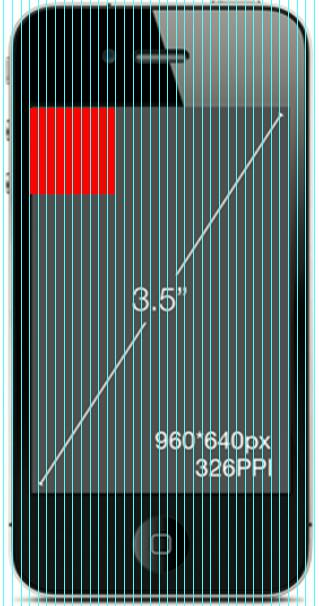
网页重构时使用的单位px，就是通常所说的像素，是网页设计中使用最多的长度单位。将显示器分成非常细小的方格，每个方格就是一个像素（这和我们理解的屏幕分辨率的1920px\*1080px的px是不同的）。不同设置下，方格的大小不一样。

未设置viewport，弹出来的都是设备的默认宽度，基本是980px，

例如iPhone4S如果不设置viewport，他就会默认是980px，就像把屏幕分成980份（不是屏幕分辨率的640px哦！）。如果设置一个元素为100px\*100px，看起来就是屏幕的100/980，可能如下图所示：



例如iPhone4S如果设置viewport width=device-width，他就会是320px，就像把屏幕分成320份（不是屏幕分辨率的640px哦！）。如果设置一个元素为100px\*100px，看起来就是屏幕的100/320，可能如下图所示：

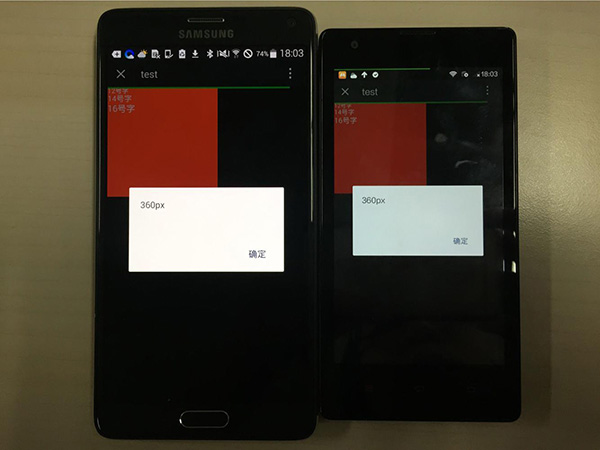
****

设置了viewport，width=device-width，弹出来的是设置好的宽度，375px、360px、320px。为什么是这个大小？这就要用到上面讲的知识点了。

iPhone6的屏幕分辨率是1334\*750px，ppi是326，所以系数是2x。那device-width就等于750/2=375px。

红米1s的屏幕分辨率是1280\*720px，ppi是312，所以系数是2x。那device-width就等于720/2=360px。

页面里的红色块给的是200\*200px，在几个设备看起来好像差不多大的样子。

上图左边是三星note4，右图是红米1s）

三星note4的屏幕分辨率是2560\*1440px，ppi是515，所以系数是4x。那device-width就等于1440/4=360px，和小米的divice-width一样大小。但是看起来左边的红色块明显大于右边，因为左边的设备大啊，虽然两者都是分成360份，但是明显左边的1份比右边的1份大

以上解释了为什么给元素赋予固定的像素值，给字体16px的大小，在大部分手机里居然能看起来差不多大小，也明白了我们并不需要写其他尺寸来适配不同的屏幕大小。

缩放

缩放与设备像素、CSS像素的关系

缩放是在放大或缩小CSS像素，比如一个宽度为 200px 的元素无论放大，还是200个CSS像素。但是因为这些像素被放大了，所以CSS像素也就跨越了更多的设备像素。缩小则相反。

缩放与视口

缩放会影响视觉视口的尺寸

页面被用户放大，视觉视口内CSS像素数量减少；被用户缩小，视觉视口内CSS像素数量增多就行了。这个道理应该是不难想的。

用户缩放不会影响布局视口

注意，这是『用户缩放』，后面会说开发者设置缩放的情况

### 缩放比例

### 

理想视口/布局视口 = device-width/layoutviewport

禁止缩放

<meta name="viewport" content="user-scalable=no">

设置缩放

<meta name="viewport" content="initial-scale=2">

使用initial-scale有一个副作用：同时也会将布局视口的尺寸设置为缩放后的尺寸。所以initial-scale=1与width=device-width的效果是一样的。

完美视口

解决各种浏览器兼容问题的理想视口设置

<meta name="viewport" content="width=device-width,initial-scale=1">

设备像素比

在谈到像素的时候，讲到除了缩放，屏幕是否为高密度也会影响设备像素和CSS像素的关系。

在缩放程度为100%（这个条件很重要，因为缩放也会影响他们）时，他们的比例叫做设备像素比(device pixel ratio)：

dpr = 设备像素 / CSS像素

可以通过JS得到： window.devicePixelRatio

设备像素比也和视口有关：

dpr = 屏幕横向设备像素 / 理想视口的宽

移动适配

移动端适配的是什么？ 我们讨论的是网页适配多种尺寸屏幕，让网页效果看起来和设计师的设计稿一样。说白了就是同一套代码在不同分辨率的手机上跑时，页面元素间的间距，留白，以及图片大小会随着变化，在比例上跟设计稿一致。