移动端基础知识

**1.四大浏览器内核**

1**.Trident (IE浏览器)** :因为在早期IE占有大量的市场份额，所以以前有很多网页是根据这个Trident的标准来编写的。

1. Gecko:( FireFox )优点就是功能强大、丰富，可以支持很多复杂网页效果和浏览器扩展接口。
2. Webkit: ( Chrome/ Safari / UC )优点就是Webkit拥有清晰的源码结构、极快的渲染速度。
3. Presto: ( 欧朋 ) Presto内核被称为公认的浏览网页速度最快的内核，同时也是处理JS脚本最兼容的内核，能在Windows、Mac及Linux操作系统下完美运行。
4. 移动端开发主要对象是手持设备，其中绝大部分是IOS和Android系统，基于Webkit内核，可使用Chrome浏览器调试即可。

**2.手机浏览器**

浏览器已经逐渐从传统桌面转向手机端，竞争也越来越激烈。目前国内市场主流的手机浏览器：UC、百度、欧朋、QQ、海豚、safari、Chrome，这些浏览器都是基于webkit内核的，兼容性方面不存在问题，同时对html5和css3的支持很好，所以，大胆地应用html5和css3技术。

**3.终端分辨率**

http://www.umindex.com/devices/android\_models



**4.移动端开发类型**

Native App（原生APP），也就是完全使用移动设备系统语言写的客户端，iPhone iPad就是纯Object-C，安卓就是纯JAVA,这是性能最棒的开发方式，但灵活性就没下面的好。

Web App 这个就是在移动浏览器里打开的，纯HTML+CSS3+JS制作

Hybrid App（混合模式移动应用）是指介于web-app、native-app这两者之间的app，兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”



**需求**

1. 开发移动端H5页面
2. 面对不同分辨率的手机
3. 面对不同屏幕尺寸的手机

视觉稿篇

在前端开发之前，视觉MM会给我们一个psd文件，称之为视觉稿。

对于移动端开发而言，为了做到页面高清的效果，视觉稿的规范往往会遵循以下两点：

1. 首先，选取一款手机的屏幕宽高作为基准(以前是iphone4的320×480，现在更多的是iphone6的375×667)。

2. 对于retina屏幕(如: dpr=2)，为了达到高清效果，视觉稿的画布大小会是基准的2倍，也就是说像素点个数是原来的4倍（对iphone6而言：原先的375×667，就会变成750×1334）。

问题：

1. 对于dpr=2的手机，为什么画布大小×2，就可以解决高清问题？

2. 对于2倍大小的视觉稿，在具体的css编码中如何还原每一个区块的真实宽高(也就是布局问题)?

了解Web App开发的基础知识

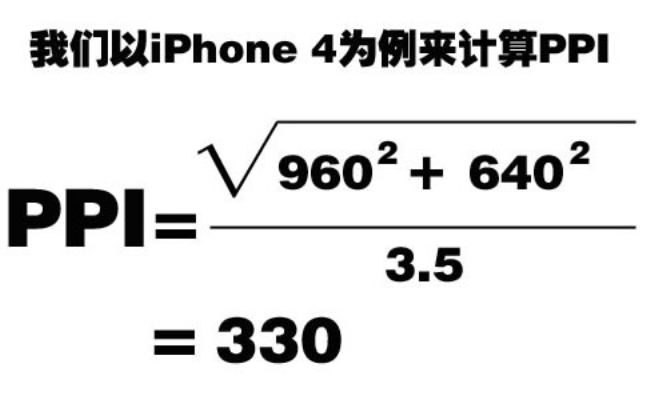
**CSS pixels：** 浏览器使用的抽象单位， 主要用来在网页上绘制内容。

**device pixels：设备无关像素** ,显示屏幕的的最小物理单位，每个dp包含自己的颜色、亮度。

等值的 CSS pixels在手机屏幕上占多大的位置，这不是固定的，这取决于很多属性。经过分析和总结，我们可以得出这么一条公式： 1 CSS pixels = （devicePixelRatio）^2 device pixels （^2是平方的意思）

**PPI/DPI**

PPI，有时也叫DPI，就是每英寸的像素数，也叫做屏幕密度,这个值越大，屏幕就越清晰。例如以HTC G7为例，480\*800的分辨率，3.7英寸，算出来就是252的PPI。



**设备像素比(device pixel ratio )**

直接翻译的话貌似叫 密度。常见取值 1.5 ， 1.0 。和标准dpi的比例（160px/inc）

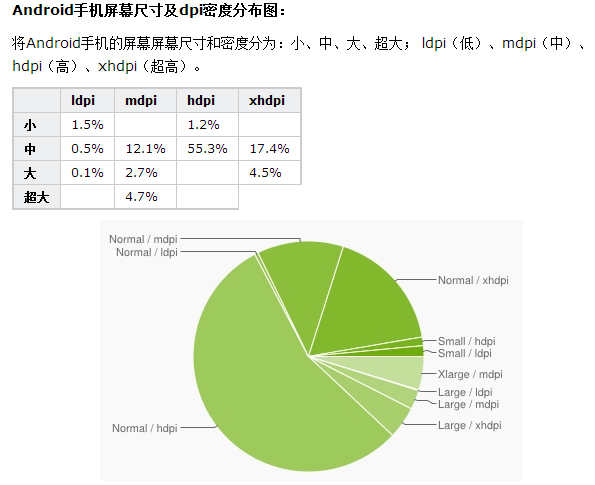
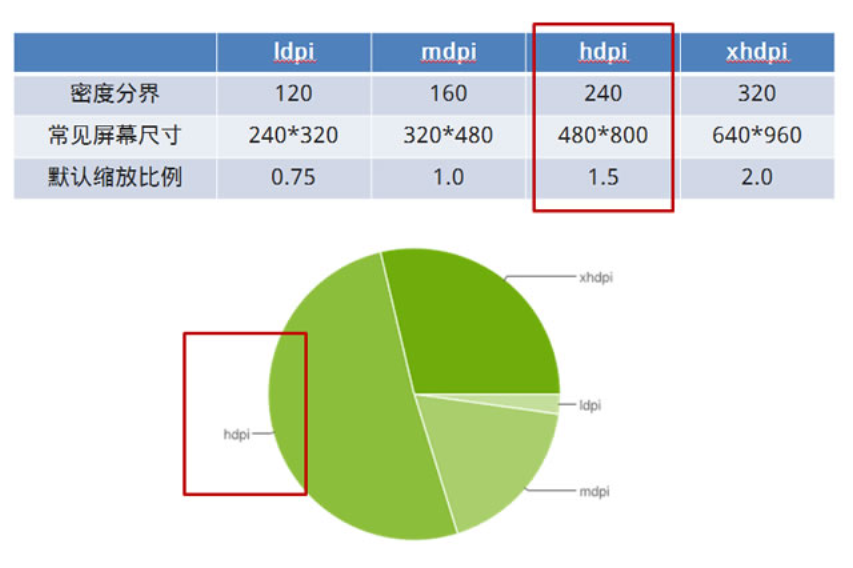
手机设备像素比如下:http://bjango.com/articles/min-device-pixel-ratio/

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

在css中,-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(这里只针对webkit内核的浏览器)

**密度决定比例**

**我们计算PPI就是为了知道一部手机设备是属于哪个密度区间的，因为不同的密度区间，对应着不同的默认缩放比例，这是一个很重要的概念。**标准dpi = 160



PPI在120-160之间的手机被归为低密度手机，160-240被归为中密度，240-320被归为高密度，320以上被归为超高密度（Apple给了它一个上流的名字——retina）

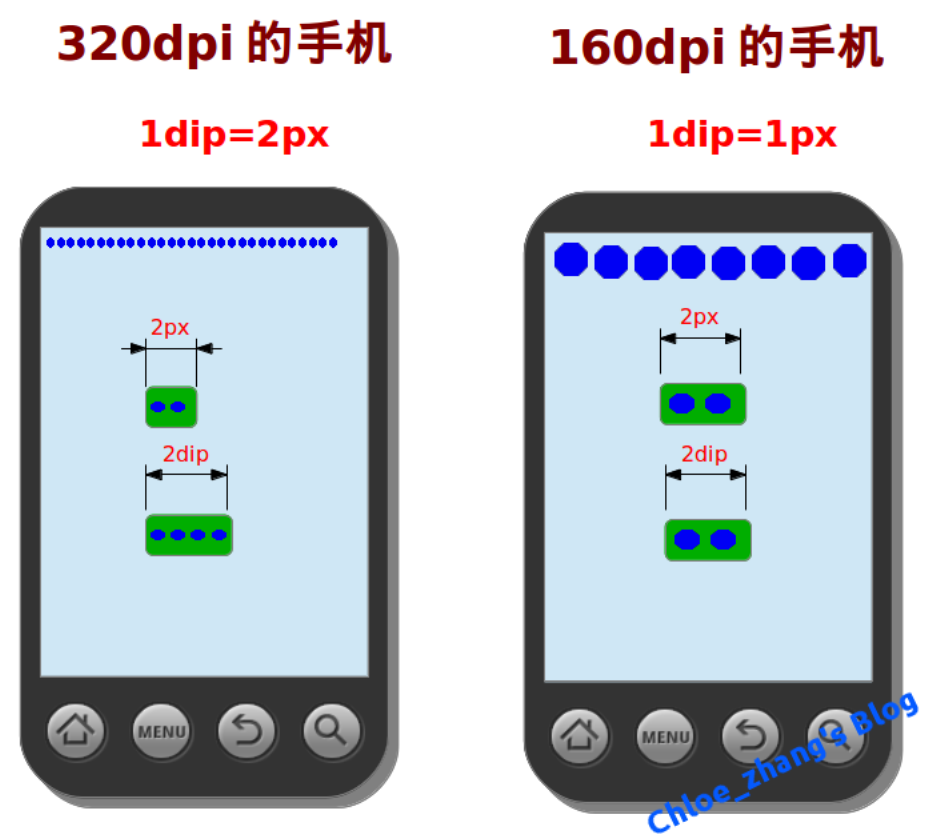
说明:

 　 是因为这个世界上存在着很多不同屏幕密度的手机，屏幕密度是什么？就是dpi，就是单位长度里的像素数量。

　　想象一下，如果这些手机的尺寸一样，屏幕密度相差很大，那么是不是说一个手机水平方向上像素很少，另一个手机水平方向上像素很多？那我们画同样pix数量的时候，它显示的长度不就会不一样了？

　　比如下面图中的两个手机，同时设置2px长度的Button，在屏幕密度较高的手机里就会显示的比较小。

　　而同时设置的2dip长度的Button，在两个手机上显示的大小是一样的。



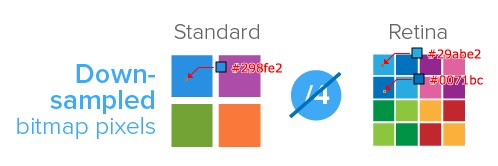
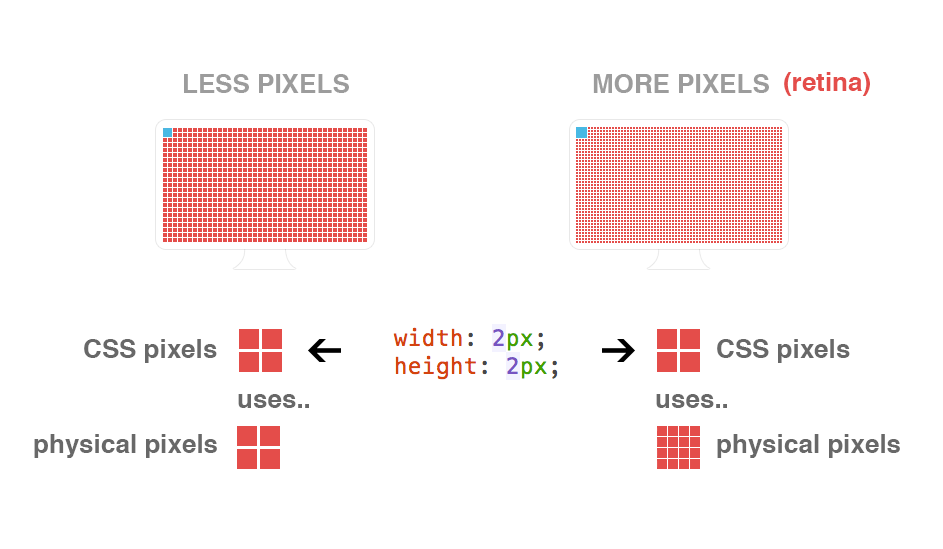
**再举个例子**

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，其物理像素就应该×2，为750×1334。

用一张图来表现，



用html5开发移动应用时往往会遇到手机的分辨率或屏幕大小不同的问题，如何使我们开发出来的应用或页面大小能适合各种手机呢?html5中的viewport能帮你做到。

　 viewport，翻译为中文可以叫做“视区”，大家都知道移动设备的屏幕一般都比PC小很多，webkit浏览器会将一个较大的“虚拟”窗口映射到移动设备的屏幕上，默认的虚拟窗口为980像素宽（目前大部分网站的标准宽度），然后按一定的比例（3：1或2：1）进行缩放。也就是说当我们加载一个普通网页的时候，webkit会先以980像素的浏览器标准加载网页，然后再缩小为490像素的宽度。注意这个缩小是一个全局缩小，也就是页面上的所有元素都会缩小。他有以下几个属性:

• width 控制 viewport 的大小，一般为了自适应设置为device-width(device-width将自动检测移动设备的屏幕宽度。所有页面内容都和移动设备屏幕自适应。)

• height 一般为了自适应设置为device-height

• target-densitydpi 设备的默认缩

◦ low-dpi – 使用ldpi作为目标 dpi。中等像素密度和高像素密度设备相应放大

◦ medium-dpi – 使用mdpi作为目标 dpi。 高像素密度设备相应放大， 像素密度设备相应缩小。 这是默认的target density.

◦ device-dpi 使用设备原本的 dpi 作为目标 dp。 不会发生默认缩放。

◦ high-dpi 使用hdpi 作为目标 dpi。 中等像素密度和低像素密度设备相应缩小

• initial-scale 初始缩放，如果你设置初始缩放为“1.0”，那么，web页面在展现的时候就会以target density分辨率的1:1来展现

• maximum-scale 最大缩放。即允许的最大缩放程度。这也是一个浮点值，用以指出页面大小与屏幕大小相比的最大乘数。

• user-scalable 用户调整缩放。即用户是否能改变页面缩放程度。如果设置为yes则是允许用户对其进行改变，反之为no。

**meta基础知识**

H5页面窗口自动调整到设备宽度，并禁止用户缩放页面

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

**忽略将页面中的数字识别为电话号码**

<meta name="format-detection" content="telephone=no" />

**忽略Android平台中对邮箱地址的识别**

<meta name="format-detection" content="email=no" />

**当网站添加到主屏幕快速启动方式，可隐藏地址栏，仅针对ios的safari**

<meta name="apple-mobile-web-app-capable" content="yes" />

<!-- ios7.0版本以后，safari上已看不到效果 -->

**将网站添加到主屏幕快速启动方式，仅针对ios的safari顶端状态条的样式**

<meta name="apple-mobile-web-app-status-bar-style" content="black" />

<!-- 可选default、black、black-translucent -->

**常见问题**

**移动端如何定义字体font-family**

中文字体使用系统默认即可，英文用Helvetica

/\* 移动端定义字体的代码 \*/

body{font-family:Helvetica;}

参考《移动端使用字体的思考》

**移动端字体单位font-size选择px还是rem**

对于只需要适配手机设备，使用px即可

对于需要适配各种移动设备，使用rem，例如只需要适配iPhone和iPad等分辨率差别比较挺大的设备