**CSS像素**

CSS像素是没有实际大小的抽象单位，它是一个相对长度，它相对的是电脑或手机显示器的分辨率。

* 1
* 2

但在不同的设备或不同的环境中，1px所代表的大小是不同的,如下表：

| **手机类型** | **华为荣耀7** | **IPhone6** |
| --- | --- | --- |
| 主屏尺寸 | 5.2英寸 | 5英寸 |
| 分辨率 | 1920PX\*1080PX | 1920PX\*1080PX |
| PPI | 424PPI | 469PPI |

同样是一英寸，同样都是像素，为啥苹果手机一英寸有469个像素，华为只有424个像素

原因请见：<http://blog.csdn.net/aiolos1111/article/details/51967744> ‘设备像素的相对性’ 

结论：   
 

**设备物理像素**

在同一个设备上，它的物理像素个数是固定的，这是厂商在出厂时就设置好了的————即一个设备的分辨率是固定的

* 1
* 2

物理像素与 dip 的比例即为网页的缩放比例，如果网页没有缩放，那么一个物理像素就对应一个 dip（设备逻辑像素）;

**DIP&DP&PT&SP 设备无关像素(逻辑像素)**

随着移动设备屏幕PPI的不断提高，对于开发者来说以前用物理像素(Physical Pixel)来度量显示元素的方法已经不奏效了。为了解决这个问题，两大平台都提出了抽象像素的概念：iOS叫做PT（Point，显示点），Android中叫做DP/DiP（Device independent Pixel，设备无关像素），如果没有特殊说明，以下统一用DP来进行描述。

SP（Scale-independent pixel）是缩放无关的像素，与DP和PT一样都是抽象像素，只不过用于描述字体的大小。

* 1
* 2
* 3

结论：

**逻辑像素和物理像素的对应关系**

IPhone手机逻辑像素与物理像素的对应关系如下图：   


Android手机逻辑像素与物理像素的对应关系如下表：

| **Screen type** | **Physic pixel** | **Logic point** |
| --- | --- | --- |
| 720p | 720\*1280 | 360\*640 |
| 1080P | 1080\*1920 | 360\*640 |
| 2k | 1440\*2560 | 360\*640 |

**dpr 设备像素缩放比**

dpr(device pixel ratio 设备像素缩放比)官方的定义为：设备物理像素和设备独立像素的比例，即：devicePixelRatio = 物理像素 / 独立像素。

* 1
* 2

So my conclusions on the mobile side are:   
1、devicePixelRatio is mostly trustworthy on most browsers.   
2、On iOS devices, multiply devicePixelRatio by screen.width to get the physical pixel count.   
3、On Android and Windows Phone devices, divide screen.width by devicePixelRatio to get the dips count.

译文：

在移动端我有如下结论

1、dpr在多数浏览器上还是非常可信的

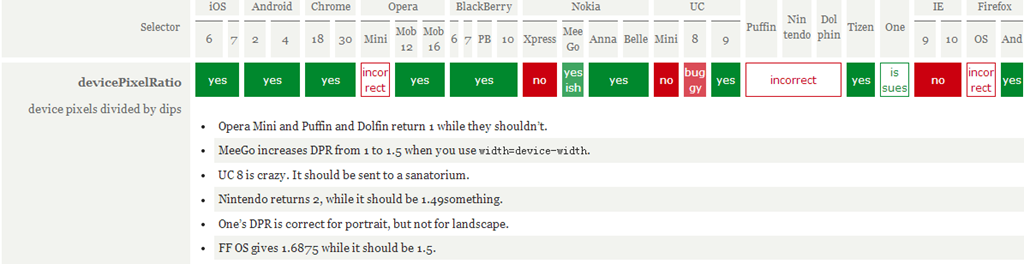
2、在IOS设备上，dpr\*screen.width = physical pixel

3、在安卓和WP设备上，dip = screen.width / dpr

4、出现2和3的不同计算公式的根源在于，不同设备的screen.width读取出的值的含义不同，具体如下：物理像素 逻辑像素dip 章节

* 1
* 2
* 3
* 4
* 5
* 6

在移动端浏览器中以及某些桌面浏览器中，window对象都有devicePixelRatio这个属性，但要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题(如下图)，所以我们现在还并不能完全信赖这个东西。



更多关于dpr的详情内容，请阅读PPK大神的：[devicePixelRatio](http://www.quirksmode.org/blog/archives/2012/06/devicepixelrati.html" \t "_blank)

原文来源：PPK大仙的[More about devicePixelRatio](http://www.quirksmode.org/blog/archives/2012/07/more_about_devi.html" \t "_blank)

**物理像素**

On iOS Retina devices, screen.width gives the width in dips. So both a retina and a non-retina iPad report 768 in portrait mode.   
On the three Android devices, screen.width gives the width in physical pixels; 480, 720, and 800, respectively. All browsers on the devices use the same values. (Imagine if some browsers on the same device used dips and others physical pixels!)

译文：   
在iOS设备上，screen.width的值为dip   
在安卓以及Windows Phone设备上，screen.width的值为物理像素

**逻辑像素dip**

Give your page a <meta name="viewport" content="width=device-width">, read out document.documentElement.clientWidth, and most browsers will give you the width of the layout viewport, which now equals the dips width.

译文：

当在页面上设置了`<meta name="viewport" content="width=device-width">`后，`document.documentElement.clientWidth`在大部分浏览器下，得到的是布局视区(layout viewport)的宽度，就等同于dip

* 1
* 2

**iOS中处理PPI**

首先规定在多高的PPI下1DP等于1px，并以这个PPI作为基准（1x multiplier），如果显示设备的PPI是基准PPI的2倍，那么1DP等于2px（2x multiplier），其实就是简单的小学乘法。

原文网址：<http://weizhifeng.net/you-should-know-about-dpi.html>

规定：

在iPhone系列中，3G/S为1x multiplier(163PPI)，其他为2x multiplier,Iphone6及以上为3x multiplier

在iPad系列中，iPad 1代和2代为1x multiplier(132PPI)，其他为2x multiplier

在iPad Mini系列中，iPad Mini一代为1x multiplier(163PPI)，其他为2x multiplier

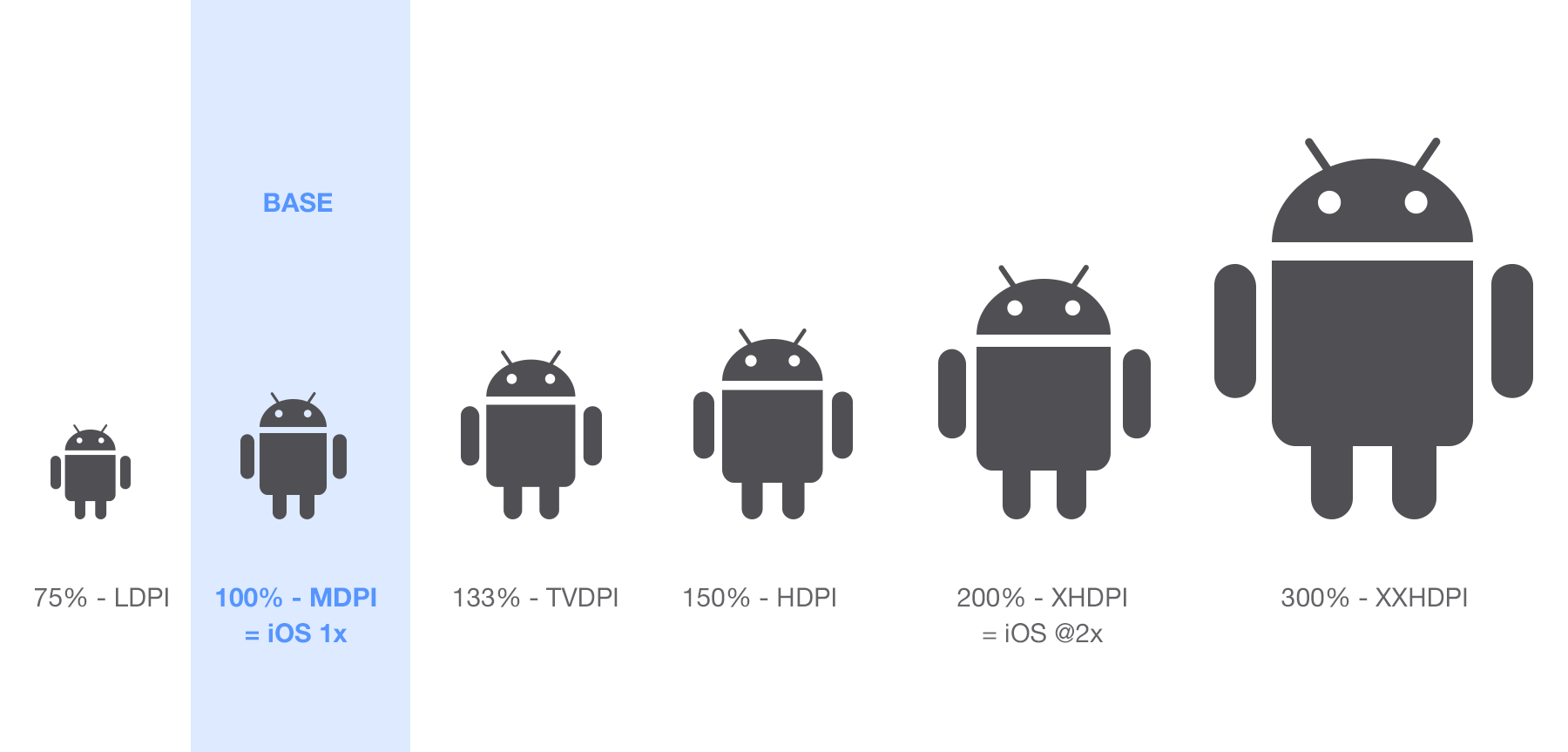
* 1
* 2
* 3
* 4
* 5

**Android处理PPI**

Android系统定义的四种像素密度

| **像素密度** | **ldpi** | **mdpi** | **hdpi** | **xhdpi** |
| --- | --- | --- | --- | --- |
| ppi | 120 | 160 | 240 | 320 |
| dpr | 0.75 | 1.0 | 1.5 | 2.0 |

可以看出Android中mdpi与iOS中的1x multiplier所代表的PPI是一样的，xhdpi与iOS的2x multiplier所代表的PPI一样，如下图



**设计页面**

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

* 1
* 2

**iPhone**

目前最多的是iPhone5/5s的屏幕。倍率为2，逻辑像素320×568。上升势头最猛，未来有望登上第一的是iPhone 6的屏幕。倍率为2，逻辑像素375×667。

按照这两种尺寸来设计，都是比较主流的做法。可以兼顾短一些的iPhone 4s，大一点的6 plus也不会过于空旷。

不过在切图的时候要注意，由于iPhone 6 plus的3倍图是由2倍图放大而来，所以位图要注意保证清晰。

* 1
* 2
* 3
* 4

**Android**

如今的Android屏幕逻辑像素已经趋于统一了：360×640

* 1
* 2

**手机端Web网页**

手机端网页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320×568。

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414×736。

* 1
* 2
* 3
* 4

**总结**

真正决定显示效果的，是逻辑像素尺寸。为此，iOS和Android平台都定义了各自的逻辑像素单位。

设计移动版web页面时，手机端网页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320×568, 即设计的页面大小为：640px × 1136px

* 1
* 2
* 3
* 4

**附：css中的1px并不总等于设备的1px**

像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。   
在为桌面浏览器设计的网页中，css的1个像素往往都是对应着电脑屏幕的1个物理像素,所以我们无需对这个津津计较，但在移动设备上，必须弄明白这点。   
在早先的移动设备中，屏幕像素密度(PPI)都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。   
还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。