

设备像素比devicePixelRatio简单介绍

这篇文章发布于 2012年08月23日, 星期四, 15:56, 归类于 [mobile相关](#)。 阅读 214839 次, 今日 35 次

本文所说 `devicePixelRatio` 其实指的是 `window.devicePixelRatio`, 被所有WebKit浏览器以及Opera所支持, 随着显示器的发展, 这个属性也慢慢登上了前端技术的舞台。

本文内容大部分属于翻译性质内容, 因此, 会不那么通俗易懂。不过, 你是做手机开发的, 或是有意向的, 本文的内容如果细细读来, 还是有些收获的。

一、定义

定义如下:

`window.devicePixelRatio`是设备上物理像素和设备独立像素(device-independent pixels (dips))的比例。

公式表示就是: $\text{window.devicePixelRatio} = \text{物理像素} / \text{dips}$

dip或dp, (device independent pixels, 设备独立像素) 与屏幕密度有关。dip可以用来辅助区分视网膜设备还是非视网膜设备。

所有非视网膜屏幕的iphone在垂直的时候, 宽度为320物理像素。当你使用 `<meta name="viewport" content="width=device-width">` 的时候, 会设置视窗布局宽度 (不同于视觉区域宽度, 不放大显示情况下, 两者大小一致, 见下图) 为320px, 于是, 页面很自然地覆盖在屏幕上。



这样，非视网膜屏幕的iphone上，屏幕物理像素320像素，独立像素也是320像素，因此，`window.devicePixelRatio` 等于 1。

而对于视网膜屏幕的iphone，如iphone4s, 纵向显示的时候，屏幕物理像素640像素。同样，当用户设置 `<meta name="viewport" content="width=device-width">` 的时候，其视区宽度并不是640像素，而是320像素，这是为了有更好的阅读体验 – 更合适的文字大小。

这样，在视网膜屏幕的iphone上，屏幕物理像素640像素，独立像素还是320像素，因此，`window.devicePixelRatio` 等于 2。

二、浏览器支持

- IE以及FireFox压根不支持。可能接下来的版本会支持。
- Opera桌面浏览器时，即使是视网膜设备，返回的值也是1而不是2. 不过，这个bug在后续的版本中会修复的。
- Opera Mobile 10不支持，不过Opera Mobile 12正确支持。
- UC总是显示1，不过其viewport属性有些让人费解。
- 只有最近的Chrome浏览器才能正确实现该属性。Chrome19返回不准确的1, Chrome22可以正确返回2.
- MeeGo WebKit (Nokia N9/N950)就吓人了：当你应用了meta viewport时候（类似 `<meta name="viewport" content="width=device-width">` ），值会从1变成1.5!

真是喜忧参半。好的是Safari, Android WebKit, Chrome 22+(Android), Opera Mobile, BlackBerry WebKit, QQ, Palm WebKit, 及Dolfin都能正确实现该属性。

当然，大部分这些浏览器仍然运行在 `devicePixelRatio` 值应该为 1 的系统上，当它们移动到视网膜类似设备时候，可能就会遇到问题。

两个注意事项：

MeeGo WebKit `meta viewport` 应用时改变值的做法是大错特错的。设备像素比应该是不变的，不仅物理像素值，设备独立像素也是如此。

二是，一些浏览器习惯在 `meta viewport` 应用时改变各种东西（三星的Dolfin就是代表），这完全就是在瞎搞。唯一的变化应该是布局视图的尺寸。如果浏览器变了其他什么都是，那都是很挫的。

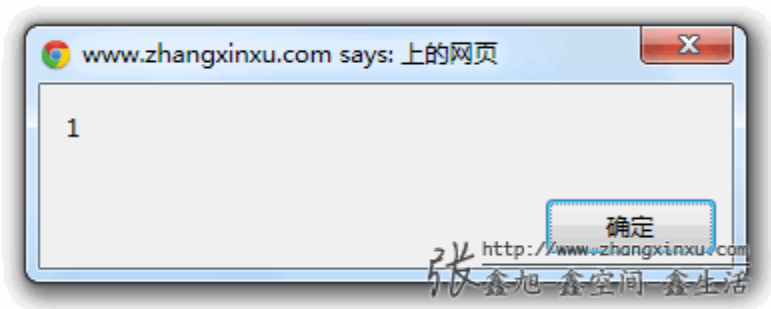
实际测试

您可以狠狠地点击这里：[window.devicePixelRatio值支持与否测试demo](#)

例如，我现在的FireFox桌面版(14.0.1)弹出的就是 `undefined`，如下图：



Chrome下是认识这个属性的，在我机子上弹出的是 `1`，如下图：



其他一些系统、设备

1. iOS

类似的，无视网膜设备 `devicePixelRatio` 值为 `1`，视网膜设备为 `2`。因为实际的像素个数是双倍。不过，iphone似乎不愿意改变大家都熟知习惯的320像素宽度布局，没有把设备宽度一下子变成640像素，因此，dips宽度依然是320，于是 `devicePixelRatio` 就是 $640/320 = 2$ 。

iOS上的情况要相对简单些，除了 `1` 就是 `2`。在其他平台也基本上很简单，因为一般分辨率都比较挫，`devicePixelRatio` 都是 `1`。

2. Android

据我所知，谷歌的Nexus One是第一个使用dips的，比iphone还早。同时Galaxy Nexus以及Galaxy Note都是类运动视网膜显示器。近距离探究这三个设备应该会有所收获。

Nexus One分辨率是480*800，为了最优的页面浏览，Android WebKit团队决定纵向手持时候的宽度依然是320像素，因此，`devicePixelRatio` 值为 $480/320 = 1.5$ 。

在同一手机上，Opera Mobile有相同的结论，dips为320宽，`devicePixelRatio` 也是 `1.5`。

顺便提一下，BlackBerry Torch 9810(OS7)物理像素同样480像素，BlackBerry WebKit团队决定坚持 `devicePixelRatio` 为 `1`。这可能是更不错的做法，在Torch显示器上480px宽度站点或多或少有些难以阅读。

Galaxy Nexus有像素的提升，为720*1200。Android团队决定提高dips层的宽度到360像素。从而使 `devicePixelRatio` 为 $720/360 = 2$ 。Chrome团队决定跟进，就如腾讯QQ浏览器所做的那样。

然而，Opera，坚持自我，dips宽度为320px，于是 `devicePixelRatio` 为 $720/320 = 2.25$ 。不过似乎还与zoom缩放层级有关。

Galaxy Note物理像素为800×1200。这里所有浏览器都决定使用与Galaxy Nexus一样的比率：Android WebKit, Chrome, 以及QQ都是 2，也就意味着其dips宽度为400px。然而，Opera依然一意孤行 2.25，于是其dips宽度值有些怪怪的：356px。

Android标准似乎不严格，于是自家人玩自家人的游戏，对于开发者而言，可能又会面临苦逼~~

3. 视网膜MacBook

新的MacBook采用视网膜显示屏，其 `devicePixelRatio` 是（如果不出意外）2。视网膜MacBook的物理像素是2800×1800，而显示出分辨率为1400×900，如果把分辨率作为dips层，则 `devicePixelRatio` 为 2 应该是无误的。

需要指出的是，如果你把分辨率改成1920×1200，`devicePixelRatio` 依然是 2。严格来讲，这是不准确的，应该是 1.5，然而你也可以说MacBook的分辨率不同于dips层，这种情况下 `devicePixelRatio` 在台式机/笔记本下的定义就不一样（哪一个？不知道。）。

在任何情况下，根据苹果的规范做法，`devicePixelRatio` 值只可能是 1 或者 2。如果你看到 2，你要提供视网膜优化显示图片，如果是 1，使用正常的图片——（这里内容其实属于视网膜站点的开发内容）。

四、其他相关属性

当页面设置了 `<meta name="viewport" content="width=device-width">` 时候，`document.documentElement.clientWidth` 在大部分浏览器下，得到的是布局视区的宽度，等同于dips的宽度。

对于 `screen.width` 的值：

- 在iOS视网膜设备上，`screen.width` 返回dips宽。因此，在竖着显示的时候，视网膜显示屏的ipad和非视网膜显示屏的ipad返回的都是768。
- 在上面提到的三个Android设备上，`screen.width` 返回的是物理像素宽度，分别480, 720, 和800。该设备上的所有浏览器都是该值。

Vasilis有一个很好的理论：苹果像素，因为它想使显示更清晰，更流畅，而Android厂商增加的像素在屏幕上塞进更多的东西。它解释了为什么苹果强调非视网膜视网膜的连续性，而Android集中在原始像素数。

Nokia Lumia Windows Phone上的IE9 `screen.width` 的值与Android设备一样，返回的是物理像素。而且其不支持 `devicePixelRatio`。因此，我们无法从中看出其对待像素的态度是如何的。

小小结论

- `devicePixelRatio` 在大多数浏览器是值得信赖的。

2. 在iOS设备, `screen.width` 乘以 `devicePixelRatio` 得到的是物理像素值。
3. 在Android以及Windows Phone设备, `screen.width` 除以 `devicePixelRatio` 得到的是设备独立像素(dips)值。

注: 本文的DIPs切勿和DPI搞混了! DPI指每英寸点的个数, 本文的DIPs指设备独立像素。

参考文章:

[devicePixelRatio](#)

[More about devicePixelRatio](#)

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。

« [SVG特征、支持以及一些实际使用问题](#)

[近期手机网页项目一些杂碎心得分享](#) »

赞助商推荐(我也要赞助):

1. **newpost**说道:
2017年01月20日 14:28
这么多设置,浏览器厂商怎么做到全部完美支持呢?
[回复](#)
2. **尹君**说道:
2016年11月3日 11:13
iPhone 6 plus 的 `window.devicePixelRatio` 是 3 了
[回复](#)
3. **gameboy**说道:
2015年12月25日 23:09
博主的第一章配图错了。这张图应该是没有设置标签的content时的样式, 如果设置了`width=device-width`,或者`initial-scale=1`的话应该会是放大的原网站左上角的。`width=device-width`缩小了layout viewport的宽度, 所以元素应该会等比例放大。
[回复](#)
4. **刘东奇**说道:
2015年10月13日 14:34
博主这里的dips按我的理解就是逻辑像素、css像素、就是web开发人员关心的像素。
[回复](#)
5. **袁源**说道:
2015年09月30日 16:09