

# 移动前端开发之viewport的深入理解

在移动设备上进行网页的重构或开发，首先得搞明白的就是移动设备上的viewport了，只有明白了viewport的概念以及弄清楚了跟viewport有关的meta标签的使用，才能更好地让我们的网页适配或响应各种不同分辨率的移动设备。

## 一、viewport的概念

通俗的讲，移动设备上的viewport就是设备的屏幕上能用来显示我们的网页的那一块区域，在具体一点，就是浏览器上(也可能是一个app中的webview)用来显示网页的那部分区域，但viewport又不局限于浏览器可视区域的大小，它可能比浏览器的可视区域要大，也可能比浏览器的可视区域要小。在默认情况下，一般来讲，移动设备上的viewport都是要大于浏览器可视区域的，这是因为考虑到移动设备的分辨率相对于桌面电脑来说都比较小，所以为了能在移动设备上正常显示那些传统的为桌面浏览器设计的网站，移动设备上的浏览器都会把自己默认的viewport设为980px或1024px（也可能是其它值，这个是由设备自己决定的），但带来的后果就是浏览器会出现横向滚动条，因为浏览器可视区域的宽度是比这个默认的viewport的宽度要小的。下图列出了一些设备上浏览器的默认viewport的宽度。

iPhone	iPad	Android Samsung	Android HTC	Chrome	Opera Presto	BlackBerry	IE
980	980	980	980	980	980	1024	1024

## 二、css中的1px并不等于设备的1px

在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的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，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。关于这点，在文章后面的部分还会讲到。

在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是  $devicePixelRatio = \text{物理像素} / \text{独立像素}$ 。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下[这篇文章](#)。

devicePixelRatio的测试结果：

Selector	iOS		Android		Chrome		Opera		BlackBerry				Nokia			UC			Puffin	Nin tendo	Dol phin	Tizen	One	IE		Firefox		
	6	7	2	4	18	30	Mini	Mob 12	Mob 16	6	7	PB	10	Xpress	Mee Go	Anna	Belle	Mini						8	9	9	10	OS
<b>devicePixelRatio</b> device pixels divided by dips	yes		yes		yes		incorrect	yes		yes		yes		no	yes ish	yes		no	bug gy	yes		incorrect		yes	is sues	no	incorrect	yes
<ul style="list-style-type: none"><li>Opera Mini and Puffin and Dolfin return 1 while they shouldn't.</li><li>MeeGo increases DPR from 1 to 1.5 when you use width=device-width.</li><li>UC 8 is crazy. It should be sent to a sanatorium.</li><li>Nintendo returns 2, while it should be 1.49something.</li><li>One's DPR is correct for portrait, but not for landscape.</li><li>FF OS gives 1.6875 while it should be 1.5.</li></ul>																												

### 三、PPK的关于三个viewport的理论

[ppk大神](#)对于移动设备上的viewport有着非常多的研究（[第一篇](#)，[第二篇](#)，[第三篇](#)），有兴趣的同学可以去看一下，本文中有很多数据和观点也是出自那里。ppk认为，移动设备上有三个viewport。

首先，移动设备上的浏览器认为自己必须能让所有的网站都正常显示，即使是那些不是为移动设备设计的网站。但如果以浏览器的可视区域作为viewport的话，因为移动设备的屏幕都不是很宽，所以那些为桌面浏览器设计的网站放到移动设备上显示时，必然会因为移动设备的viewport太窄，而挤作一团，甚至布局什么的都会乱掉。也许有人会问，现在不是有很多手机分辨率都非常大吗，比如768x1024，或者1080x1920这样，那这样的手机用来显示为桌面浏览器设计的网站是没问题的吧？前面我们已经说了，css中的1px并不是代表屏幕上的1px，你分辨率越大，css中1px代表的物理像素就会越多，devicePixelRatio的值也越大，这很好理解，因为你分辨率增大了，但屏幕尺寸并没有变大多少，必须让css中的1px代表更多的物理像素，才能让1px的东西在屏幕上的大小与那些低分辨率的设备差不多，不然就会因为太小而看不清。所以在1080x1920这样的设备上，在默认情况下，也许你只要把一个div的宽度设为300多px（视devicePixelRatio的值而定），就是满屏的宽度了。回到正题上来，如果把移动设备上浏览器的可视区域设为viewport的话，某些网站就会因为viewport太窄而显示错乱，所以这些浏览器就决定默认情况下把viewport设为一个较宽的值，比如980px，这样的话即使是那些为桌面设计的网站也能在移动浏览器上正常显示了。ppk把这个浏览器默认的viewport叫做 *layout viewport*。这个layout viewport的宽度可以通过document.documentElement.clientWidth 来获取。

然而，*layout viewport* 的宽度是大于浏览器可视区域的宽度的，所以我们还需要一个viewport来代表 浏览器可视区域的大小，ppk把这个viewport叫做 *visual viewport*。visual viewport的宽度可以通过window.innerWidth 来获取，但在Android 2, Opera mini 和 UC 8中无法正确获取。





现在我们已经有两个viewport了：*layout viewport* 和 *visual viewport*。但浏览器觉得还不够，因为现在越来越多的网站都会为移动设备进行单独的设计，所以必须还要有一个能完美适配移动设备的viewport。所谓的完美适配指的是，首先不需要用户缩放和横向滚动条就能正常的查看网站的所有内容；第二，显示的文字的大小是合适，比如一段14px大小的文字，不会因为

在一个高密度像素的屏幕里显示得太小而无法看清，理想的情况是这段14px的文字无论是在何种密度屏幕，何种分辨率下，显示出来的大小都是差不多的。当然，不只是文字，其他元素像图片什么的也是这个道理。ppk把这个viewport叫做 *ideal viewport*，也就是第三个viewport——移动设备的理想viewport。

*ideal viewport*并没有一个固定的尺寸，不同的设备拥有有不同的*ideal viewport*。所有的iphone的*ideal viewport*宽度都是320px，无论它的屏幕宽度是320还是640，也就是说，在iphone中，css中的320px就代表iphone屏幕的宽度。



但是安卓设备就比较复杂了，有320px的，有360px的，有384px的等等，关于不同的设备*ideal viewport*的宽度都为多少，可以到<http://viewportsizes.com>去查看一下，里面收集了众多设备的理想宽度。

再总结一下：ppk把移动设备上的viewport分为*layout viewport*、*visual viewport*和*ideal viewport*三类，其中的*ideal viewport*是最适合移动设备的viewport，*ideal viewport*的宽度等于移动设备的屏幕宽度，只要在css中把某一元素的宽度设为*ideal viewport*的宽度(单位用px)，那么这个元素的宽度就是设备屏幕的宽度了，也就是宽度为100%的效果。*ideal viewport*的意义在于，无论在何种分辨率的屏幕下，那些针对*ideal viewport*而设计的网站，不需要用户手动缩放，也不需要出现横向滚动条，都可以完美的呈现给用户。

#### 四、利用meta标签对viewport进行控制

移动设备默认的viewport是*layout viewport*，也就是那个比屏幕要宽的viewport，但在进行移动设备网站的开发时，我们需要的是*ideal viewport*。

那么怎么才能得到ideal viewport呢？这就该轮到meta标签出场了。

我们在开发移动设备的网站时，最常见的的一个动作就是把下面这个东西复制到我们的head标签中：

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

该meta标签的作用是让当前viewport的宽度等于设备的宽度，同时不允许用户手动缩放。也许允不允许用户缩放不同的网站有不同的要求，但让viewport的宽度等于设备的宽度，这个应该是大家都想要的效果，如果你不这样的设定的话，那就会使用那个比屏幕宽的默认viewport，也就是说会出现横向滚动条。

这个name为viewport的meta标签到底有哪些东西呢，又都有什么作用呢？

meta viewport 标签首先是由苹果公司在其safari浏览器中引入的，目的就是解决移动设备的viewport问题。后来安卓以及各大浏览器厂商也都纷纷效仿，引入对meta viewport的支持，事实也证明这个东西还是非常有用的。

在苹果的规范中，meta viewport 有6个属性(暂且把content中的那些东西称为一个个属性和值)，如下：

width	设置 <i>layout viewport</i> 的宽度，为一个正整数，或字符串"width=device-width"
initial-scale	设置页面的初始缩放值，为一个数字，可以带小数
minimum-scale	允许用户的最小缩放值，为一个数字，可以带小数
maximum-scale	允许用户的最大缩放值，为一个数字，可以带小数
height	设置 <i>layout viewport</i> 的高度，这个属性对我们并不重要，很少使用
user-scalable	是否允许用户进行缩放，值为"no"或"yes", no 代表不允许，yes 代表允许

这些属性可以同时使用，也可以单独使用或混合使用，多个属性同时使用时用逗号隔开就行了。

此外，在安卓中还支持 target-densitydpi 这个私有属性，它表示目标设备的密度等级，作用是决定css中的1px代表多少物理像素

target-densitydpi	值可以为一个数值或 high-dpi 、 medium-dpi、 low-dpi、 device-dpi
-------------------	--



特别说明的是，当 `target-densitydpi=device-dpi` 时，css中的1px会等于物理像素中的1px。

因为这个属性只有安卓支持，并且安卓已经决定要废弃~~target-densitydpi~~ 这个属性了，所以这个属性我们要避免进行使用 。

## 五、把当前的viewport宽度设置为 ideal viewport 的宽度

要得到ideal viewport就必须把默认的layout viewport的宽度设为移动设备的屏幕宽度。因为meta viewport中的width能控制layout viewport的宽度，所以我们只需要把width设为width-device这个特殊的值就行了。

```
<meta name="viewport" content="width=device-width">
```

下图是这句代码在各大移动端浏览器上的测试结果：

Test	iPhone	iPad	Android Samsung	Android HTC	Chrome	Opera Presto	BlackBerry	IE
<a href="#">width=device-width</a> Does width=device-width give the layout viewport the width of the ideal viewport?	portrait	portrait	yes	yes	yes	yes	yes	yes
• Safari applies the portrait width (320 or 768) to both landscape and portrait modes.								

可以看到通过width=device-width，所有浏览器都能把当前的viewport宽度变成ideal viewport的宽度，但要注意的是，在iphone和ipad上，无论是竖屏还是横屏，宽度都是竖屏时ideal viewport的宽度。

这样的写法看起来谁都会做，没吃过猪肉，谁还没见过猪跑啊~，确实，我们在开发移动设备上的网页时，不管你明不明白什么是viewport，可能你只需要这么一句代码就够了。

可是你肯定不知道

```
<meta name="viewport" content="initial-scale=1">
```

这句代码也能达到和前一句代码一样的效果，也可以把当前的viewport变为 ideal viewport。

呵呵，傻眼了吧，因为从理论上讲，这句代码的作用只是不对当前的页面进行缩放，也就是页面本该是多大就是多大。那为什么会有 width=device-width 的效果呢？

要想清楚这件事情，首先你得弄明白这个缩放是相对于什么来缩放的，因为这里的缩放值是1，也就是没缩放，但却达到了 ideal viewport 的效果，所以，那答案就只有一个了，缩放是相对于 ideal viewport来进行缩放的，当对 ideal viewport进行100%的缩放，也就是缩放值为1的时候，不就得到了 ideal viewport吗？事实证明，的确是这样的。下图是各大移动端的浏览器当设置了<meta name="viewport" content="initial-scale=1"> 后是否能把当前的viewport宽度变成 ideal viewport 的宽度的测试结果。

Test	iPhone	iPad	Android Samsung	Android HTC	Chrome	Opera Presto	BlackBerry	IE
<a href="#">initial-scale=1</a> Does initial-scale=1 give the layout viewport the width of the ideal viewport?	yes	yes	yes	yes	yes	yes	yes	portrait
• IE10 applies the portrait width (320) to both landscape and portrait modes.								

测试结果表明 initial-scale=1 也能把当前的viewport宽度变成 ideal viewport 的宽度，但这次轮到了windows phone 上的IE 无论是竖屏还是横屏都把宽度设为竖屏时ideal viewport的宽度。但这点小瑕疵已经无关紧要了。

但如果width 和 initial-scale=1同时出现，并且还出现了冲突呢？比如：

```
<meta name="viewport" content="width=400, initial-scale=1">
```

width=400表示把当前viewport的宽度设为400px，initial-scale=1则表示把当前viewport的宽度设为ideal viewport的宽度，那么浏览器到底该服从哪个命令呢？是书写顺序在后面的那个吗？不是。当遇到这种情况时，浏览器会取它们两个中较大的那个值。例如，当width=400，ideal viewport的宽度为320时，取的是400；当width=400，ideal viewport的宽度为480时，取的是ideal viewport的宽度。（ps:在uc9浏览器中，当initial-scale=1时，无论width属性的值为多少，此时viewport的宽度永远都是ideal viewport的宽度）



最后，总结一下，要把当前的viewport宽度设为ideal viewport的宽度，既可以设置 width=device-width，也可以设置 initial-scale=1，但这两者各有一个小缺陷，就是iphone、ipad以及IE 会横竖屏不分，通通以竖屏的ideal viewport宽度为准。所以，最完美的写法应该是，两者都写上去，这样就 initial-scale=1 解决了 iphone、ipad的毛病，width=device-width则解决了IE的毛病：

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

## 六、关于meta viewport的更多知识

### 1、关于缩放以及initial-scale的默认值

首先我们先来讨论一下缩放的问题，前面已经提到过，缩放是相对于ideal viewport来缩放的，缩放值越大，当前viewport的宽度就会越小，反之亦然。例如在iphone中，ideal viewport的宽度是320px，如果我们设置 initial-scale=2，此时viewport的宽度会变为只有160px了，这也好理解，放大了一倍嘛，就是原来1px的东西变成2px了，但是1px变为2px并不是把原来的320px变为640px了，而是在实际宽度不变的情况下，1px变得跟原来的2px的长度一样了，所以放大2倍后原来需要320px才能填满的宽度现在只需要160px就做到了。因此，我们可以得出一个公式：

$$\text{visual viewport宽度} = \text{ideal viewport宽度} / \text{当前缩放值}$$
$$\text{当前缩放值} = \text{ideal viewport宽度} / \text{visual viewport宽度}$$

ps: visual viewport的宽度指的是浏览器可视区域的宽度。

大多数浏览器都符合这个理论，但是安卓上的原生浏览器以及IE有些问题。安卓自带的webkit浏览器只有在 initial-scale = 1 以及没有设置width属性时才是表现正常的，也就相当于这理论在它身上基本没用；而IE则根本不用 initial-scale这个属性，无论你给他设置什么，initial-scale表现出来的效果永远是1。

好了，现在再来说下initial-scale的默认值问题，就是不写这个属性的时候，它的默认值会是多少呢？很显然不会是1，因为当 initial-scale = 1 时，当前的layout viewport宽度会被设为 ideal viewport的宽度，但前面说了，各浏览器默认的 layout viewport宽度一般都是980啊，1024啊，800啊等等这些个值，没有一开始就是 ideal viewport的宽度的，所以 initial-scale的默认值肯定不是1。安卓设备上的initial-scale默认值好像没有方法能够得到，或者就是干脆它就没有默认值，一定要你显示的写出来这个东西才会起作用，我们不管它了，这里我们重点说一下iphone和ipad上的initial-scale默认值。

根据测试，我们可以在iphone和ipad上得到一个结论，就是无论你给layout viewport设置的宽度是多少，而又没有指定初始的缩放值的话，那么iphone和ipad会自动计算initial-scale这个值，以保证当前layout viewport的宽度在缩放后就是浏览器可视区域的宽度，也就是说不会出现横向滚动条。比如说，在iphone上，我们不设置任何的viewport meta标签，此时layout viewport的宽度为980px，但我们可以看到浏览器并没有出现横向滚动条，浏览器默认地把页面缩小了。根据上面的公式，当前缩放值 = ideal viewport宽度 / visual viewport宽度，我们可以得出：

$$\text{当前缩放值} = 320 / 980$$

也就是当前的initial-scale默认值应该是 0.33这样子。当你指定了initial-scale的值后，这个默认值就不起作用了。

总之记住这个结论就行了：在iphone和ipad上，无论你给viewport设的宽的是多少，如果没有指定默认的缩放值，则iphone和ipad会自动计算这个缩放值，以达到当前页面不会出现横向滚动条(或者说viewport的宽度就是屏幕的宽度)的目的。





## 2、动态改变meta viewport标签

### 第一种方法

可以使用document.write来动态输出meta viewport标签，例如：

```
document.write('<meta name="viewport" content="width=device-width,initial-s
```

### 第二种方法

通过setAttribute来改变

```
<meta id="testViewport" name="viewport" content="width = 380">
<script>
var mvp = document.getElementById('testViewport');
mvp.setAttribute('content','width=480');
</script>
```

安卓2.3自带浏览器上的一个bug



```
<meta name="viewport" content="width=device-width">
```

```
<script type="text/javascript">
```

```
alert(document.documentElement.clientWidth); //弹出600, 正常情况应该弹出320
```

```
</script>
```

```
<meta name="viewport" content="width=600">
```

```
<script type="text/javascript">
```

```
alert(document.documentElement.clientWidth); //弹出320, 正常情况应该弹出600
```

```
</script>
```

测试的手机ideal viewport 宽度为320px，第一次弹出的值是600,但这个值应该是第行meta标签的结果啊，然后第二次弹出的值是320，这才是第一行meta标签所达到的效果啊，所以在安卓2.3(或许是所有2.x版本中)的自带浏览器中，对meta viewport标签进行覆盖或更改，会出现让人非常迷糊的结果。

## 七、结语

说了那么多废话，最后还是有必要总结一点有用的出来。

首先如果不设置meta viewport标签，那么移动设备上浏览器默认的宽度值为800px，980px，1024px等这些，总之是大于屏幕宽度的。这里的宽度所用的单位px都是指css中的px，它跟代表实际屏幕物理像素的px不是一回事。

第二、每个移动设备浏览器中都有一个理想的宽度，这个理想的宽度是指css中的宽度，跟设备的物理宽度没有关系，在css中，这个宽度就相当于100%的所代表的那个宽度。我们可以用meta标签把viewport的宽度设为那个理想的宽度，如果不知道这个设备的理想宽度是多少，那么用device-width这个特殊值就行了，同时initial-scale=1也有把viewport的宽度设为理想宽度的作用。所以，我们可以使用

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

来得到一个理想的viewport（也就是前面说的ideal viewport）。

为什么需要有理想的viewport呢？比如一个分辨率为320x480的手机理想viewport的宽度是320px，而另一个屏幕尺寸相同但分辨率为640x960的手机的理想viewport宽度也是为320px，那为什么分辨率大的这个手机的理想宽度

要跟分辨率小的那个手机的理想宽度一样呢？这是因为，只有这样才能保证同样的网站在不同分辨率的设备上看起来都是一样或差不多的。实际上，现在市面上虽然有那么多不同种类不同品牌不同分辨率的手机，但它们的理想viewport宽度归纳起来无非也就 320、360、384、400等几种，都是非常接近的，理想宽度的相近也就意味着我们针对某个设备的理想viewport而做出的网站，在其他设备上的表现也不会相差非常多甚至是表现一样的。