

理解CSS像素，设备像素和设备独立像素

链接：<https://www.jianshu.com/p/5fc965d26bdb>

DP(device pixels)

设备像素也被称作为物理像素，它表示显示设备的真实像素，此像素是设备的固有属性，也就是说，从出厂的那一刻，设备像素已经固定，不会再发生改变。单位 `pt`。

分辨率

通常使用分辨率来描述设备像素，例如 `1920*1080`，表示设备横向有1920个像素点，纵向有1080个像素点。

屏幕尺寸

购买电视或者手机等设备的显示器都有一个屏幕尺寸参数，比如55英寸，它既不是显示器的宽度也不是高度，而是对角线的长度。

PPI(pixels per inch)

每英寸像素取值，更确切的说法应该是像素密度，也就是衡量单位物理面积内拥有像素值的情况。已知屏幕分辨率和主对角线的尺寸，则 `ppi` 等于以iphone 6为例：

屏幕	触摸屏类型	电容屏，多点触控
	主屏尺寸	5.5英寸
	主屏材质 ⓘ	Retina HD
	主屏分辨率 ⓘ	1920x1080像素
	屏幕像素密度	401ppi

```
var 斜边尺寸 = √(1920^2+1080^2) √代表开根号
var ppi = 斜边尺寸/5.5
ppi = 401ppi
```

$$PPI = \frac{d_p}{d_i} = \frac{\sqrt{w_p^2 + h_p^2}}{d_i}$$

其中，

- d_p 为屏幕对角线的分辨率
- w_p 为屏幕横向分辨率
- h_p 为屏幕纵向分辨率
- d_i 为屏幕对角线的长度(单位为英寸)。

我们知道，`ppi` 越高，每英寸像素点越多，图像越清晰；我们可以类比物体的密度，密度越大，单位体积的质量就越大，`ppi` 越高，单位面积的像素越多。

DIP(Device independent Pixel)

设备独立像素，也称为逻辑像素，简称 `dip`。

由来

随着技术的进步，很小的屏幕能容纳很高的物理像素，比如 `1920*1080` 以前只能应用于大的显示器，现在可能被容纳于一个小的移动端设备中。如果我们直接使用物理像素，看如下代码：

```
div {  
  width:200px;  
  height:100px;  
}
```

iPhone 3和iPhone 4，屏幕尺寸一样，后者的像素密度是前者的两倍，如果使用物理像素作为单位，那么 `div` 的宽度视觉上iPhone 4是iPhone 3的一半，所以使用物理像素在web中设置尺寸是不合理的。因此操作系统定义设备独立像素，用设备独立像素定义的尺寸，不管屏幕的参数如何，都能以合适的大小显示。iPhone 3物理像素是 `320*480`，iPhone 4的物理像素是 `640*960`，但是它们的设备独立像素都是 `320*480`，iPhone 4用4个物理像素表示1个设备独立像素（横向2个物理像素表示1个设备独立像素，纵向2个物理像素表示一个设备独立像素），这一切都是由操作系统控制。设备独立像素可以由如下代码获取：

```
screen.width //宽度  
screen.height //高度
```

PX(CSS pixels)

虚拟像素，可以理解为“直觉”像素，在web中我们所设置的像素都是CSS像素，CSS像素的单位是 `px`。在CSS规范中，长度单位可以分为两类，绝对(`absolute`)单位以及相对(`relative`)单位。`px`是一个相对单位，相对的是设备像素(`device pixel`)。

在没有缩放的情况下，1个css像素等同于一个设备独立像素。

CSS像素在视觉上是很容易改变大小的，比如缩放浏览器页面，就是改变的CSS像素，当放大一倍，那么一个CSS像素在横向或者纵向上会覆盖两个设备独立像素。例如宽度 `100px` 像素，当页面放大一倍，它会在横向上由原本占据100个设备独立像素，变成占据200个设备独立像素；如果缩小，则恰好相反，只能占据50个设备独立像素。

DPR(device pixels ratio)

设备像素比 `dpr` 描述的是未缩放状态下，设备像素和设备独立像素的比例关系。在JS中可以通过 `window.devicePixelRatio` 获取。

DPR = 设备像素/设备独立像素

理解

知道设备像素比之后，我们就可以将程序中使用到的CSS像素转换为设备像素。设备像素比(`dpr`)是指在移动开发中1个css像素占用多少设备像素，如2代表1个css像素用2x2个设备像素来绘制。设备像素比(`dpr`)，公式为 $1px = (dpr)^2 * 1dp$ ，可以理解为 `1px` 由多少个设备像素组成。

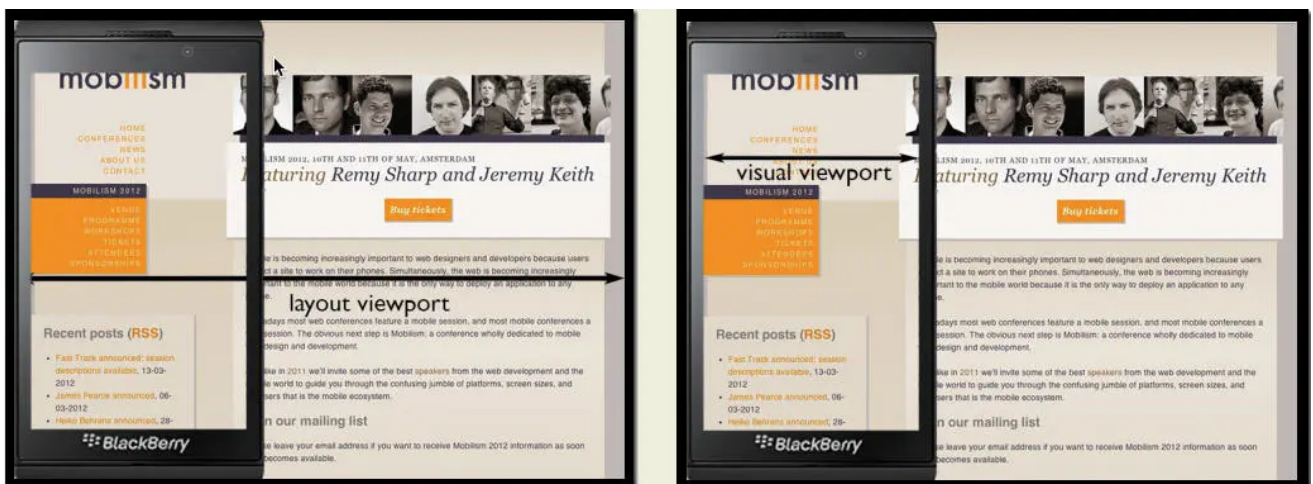
Viewport

PPK的关于三个viewport的理论

ppk大神认为，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？首先，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。

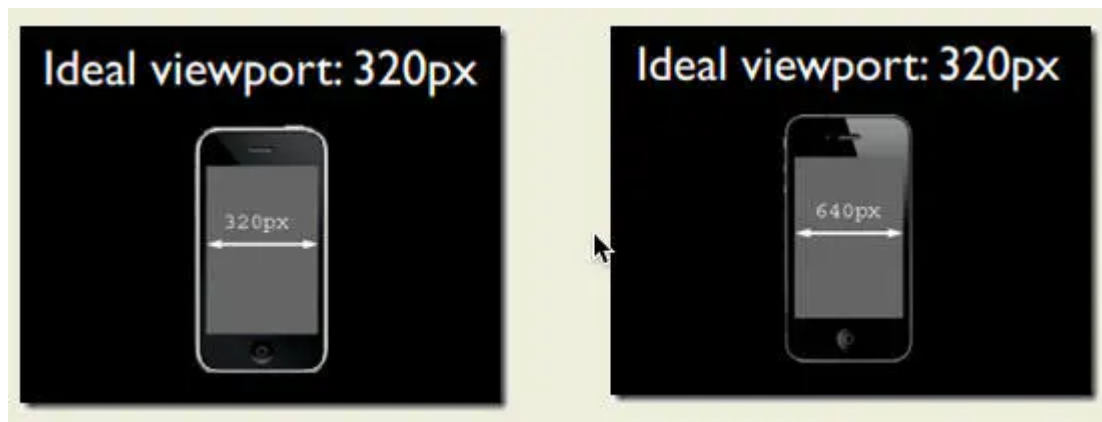
这个 layout viewport 的宽度可以通过 document.documentElement.clientWidth 来获取。

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。



现在我们已经有了两个 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。

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。

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=0">
```

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：

属性	描述
width	设置layout viewport的宽度，为一个整数，或字符串"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layout viewport的高度，这个属性很少使用
user-scalable	是否允许用户进行缩放，值为"no" 或"yes"，

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。

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 device-width 这个特殊的值就行了。

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

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。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。

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。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。

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">
```