一个移动端的时代

从我工作以来，开发的一直都是移动端的页面，只有偶尔去开发几个PC端的页面，现在是一个移动端的时代，移动先行已经深入骨髓，作为一个web前端开发，如果你还在为如何开发移动端页面而迷茫，或者你还在为开发出了一个在你手机上“完美”的移动页面而沾沾自喜却不知移动的世界有多“残酷”的时候，那你应该看看这篇文章了。希望这能给你帮助，同时也能给我帮助，有不合理的地方，欢迎评论支持，必将改正。

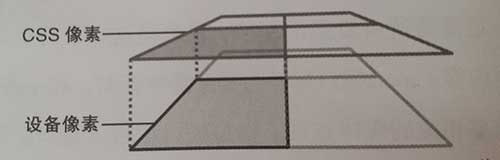


这里是基础，你了解否？

一、像素 - 什么是像素

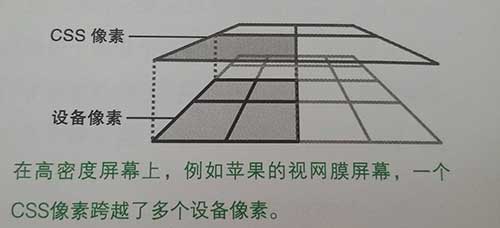
像素是web页面布局的基础，那么到底什么才是一个像素呢？  
像素：**一个像素就是计算机屏幕所能显示一种特定颜色的最小区域。** 这是像素的概念，实际上，在web前端开发领域，像素有以下两层含义：  
1、**设备像素：**设备屏幕的物理像素，对于任何设备来讲物理像素的数量是固定的。  
2、**CSS像素：**这是一个抽象的像素概念，它是为web开发者创造的。

如下图，是在缩放比例为1，即scale = 1的情况下，设备像素和CSS像素示意图

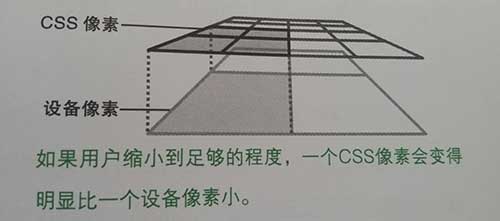


现在你已经了解了，原来像素对于web前端开发来讲有这样的两层含义，那么你有没有再深入的考虑这样一个问题，当我给一个元素设置了 **width: 200px;** 这条样式的时候，到底放生了什么事情？  
你可能会说：“废话！元素的宽度是200px呗。”；对，并没有什么问题，但是这个200px指的是什么呢？因为我们知道，对于web前端来讲像素有两层含义，那么到底是设备像素还是CSS像素？实际上我们控制的是CSS像素，因为前面提到了，CSS像素是给我们web前端开发者创造的抽象概念。所以你要记住：**当你给元素设置了 width: 200px 时，这个元素的宽度跨越了200个CSS像素**。但是它并不一定跨越200个设备像素，至于会跨越多少个设备像素，就取决于**手机屏幕的特性**和**用户的缩放**了,举个栗子：

苹果手机的视网膜屏幕，是一个高密度屏幕，它的像素密度是普通屏幕的2倍，所以当我们设置 **width: 200px;** 时，200个CSS像素跨越了400个设备像素，如下图：



如果用户缩小页面，那么一个CSS像素会明显小于一个设备像素，这个时候 **width: 200px;** 这条样式中所设置的200个CSS像素跨越不了200个设备像素，如下图：



让我们来做一个总结

1、web前端领域，像素分为**设备像素**和**CSS像素**  
3、一个CSS像素的大小是可变的，比如用后缩放页面的时候，实际上就是在缩小或放大CSS像素，而设备像素无论大小还是数量都是不变的。

二、移动端的三个视口

一看标题，你是不是蒙了？三个视口？什么三个视口？先别急，让我们慢慢来讲。  
你一定写过这样一条样式：**width: 25%;** 但是你有想过给一个元素加上这样一条样式之后发生了什么吗？25%是基于谁的25%？明白的同学可能知道了：一个块元素默认的宽度是其父元素的100%，是基于起父元素的，所以25%指的是父元素宽度的25%，所以，**body**元素的默认宽度是**html**元素宽度的100%，那么你有没有想过**html**元素的宽度是基于谁的呢？这个时候，就要引出一个概念：**初始包含块和视口了**

记住一句话：**视口是html的父元素，所以我们称视口为初始包含块。** 这样你就明白了，html元素的百分比是基于视口的。

第一个视口：布局视口

首先你需要了解一个原因：浏览器厂商是希望满足用户的要求的，即在手机也能浏览为PC端设计的网站，所以浏览器厂商必须想办法来在满足~  
在PC浏览器中，视口只有一个，并且 **视口的宽度** = **浏览器窗口的宽度**，但是在移动端也要根据这个来设计的话，那么PC端设计的网站在移动端看起来会很丑，因为PC端的网页宽度在800 ~ 1024个CSS像素，而手机屏幕要窄的多，这个时候再PC端25%的宽度在移动端看起来会很窄。所以，布局视口的概念产生了。

布局是口：移动端CSS布局的依据视口，即CSS布局会根据布局视口来计算。  
也就是说，在移动端，视口和浏览器窗口将不在关联，实际上，布局视口要比浏览器窗口大的多(在手机和平板中浏览器布局视口的宽度在768~1024像素之间)，如下图（布局视口和窗口的关系）：



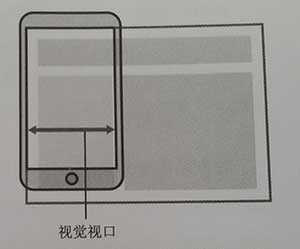
可以通过以下JavaScript代码获取布局视口的宽度和高度：

document.documentElement.clientWidth

document.documentElement.clientHeight

第二个视口：视觉视口

视觉视口可能要更好理解一些，他就是用户正在看到网站的区域，如下图：



第三个视口：理想视口

理想视口，这是我们最需要关注的视口，现在我们来回顾一下我们知道了哪些视口，有两个，分别是：布局视口，视觉视口。  
我们前面提到过，布局视口的宽度一般在 680~1024像素之间，这样可以使得PC网站在手机中不被压扁，但是这并不理想，因为手机更适合窄的网站，换句话说，布局视口并不是最理想的宽度，所以，就引入了理想视口。

理想视口，定义了理想视口的宽度，比如对于iphone5来讲，理想视口是320\*568。但是最终作用的还是布局视口，因为我们的css是依据布局视口计算的，所以你可以这样理解理想视口：**理想的布局视口**。下面这段代码可以告诉手机浏览器要把布局视口设为理想视口：

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

上面那段代码告诉浏览器：**将布局视口的宽度设为理想视口。**所以，上面代码中的width指的是布局视口的宽 device-width 实际上就是理想视口的宽度。

好了，移动端的三个视口介绍完了，让我们总结一下：

1、在PC端，布局视口就是浏览器窗口  
2、在移动端，视口被分为两个：布局视口、视觉视口。  
3、移动端还有一个理想视口，它是布局视口的理想尺寸，即理想的布局视口。（注：理想视口的尺寸因设备和浏览器的不同而不同，但这对于我们来说无所谓）  
4、可以将布局视口的宽度设为理想视口

三、设备像素比(DPR)

下面你还需要了解一个概念，设备像素比(Device Pixel Ratio 简称：DPR)。  
下面是设备像素比的计算公式

公式成立的大前提：**（缩放比例为1）**  
**设备像素比(DPR) = 设备像素个数 / 理想视口CSS像素个数(device-width)**

与理想视口一样，设备像素比对于不同的设备是不同的，但是他们都是合理的，比如早起iphone的设备像素是320px，理想视口也是320px，所以早起iphone的DPR=1，而后来iphone的设备像素为640px，理想视口还是320px，所以后来iphone的DPR=2。在开发中，打开浏览器的调试工具可以看到设备像素比，如下图：



如上图，我们可以获得以下信息：  
iPhone5的理想视口是：320\*568 <==> device-width = 320，device-height = 568  
iPhone5的设备像素比：2  
根据公式：**设备像素比(DPR) = 设备像素个数 / 理想视口CSS像素个数(device-width)**  
可知iPhone5的设备像素为 640\*1136

缩放

在讲设备像素比公式的时候讲到了：

公式成立的大前提：**（缩放比例为1）**  
**设备像素比(DPR) = 设备像素个数 / 理想视口CSS像素个数(device-width)**

那么缩放到底是什么呢？也许这个问题让你很疑惑，如果你自己阅读了前面的内容，你会注意到CSS像素的大小是可变得，而缩放从技术实现的角度来讲，就是放大或缩小CSS像素的过程，怎么样？明白了吧，当你用双指缩放页面的时候，实际上是在放大或缩小CSS像素，至于什么是CSS像素，晕，回去从头好好看

也学你会觉得缩放没什么，但是你了解这个概念至关重要，因为在《一篇真正教会你开发移动端页面的文章(二)》中，会用到这里的概念。即

缩放：缩小放大的是 CSS像素。

meta标签

meta视口标签存在的主要目的是为了让布局视口和理想视口的宽度匹配，meta视口标签应该放在HTML文档的head标签内，语法如下：

<meta name="viewport" content="name=value,name=value" />

其中 content 属性是一个字符串值，字符串是由逗号“，”分隔的 名/值 对组成，共有5个：

1、width：设置布局视口的宽  
2、init-scale：设置页面的初始缩放程度  
3、minimum-scale：设置了页面最小缩放程度  
4、maximum-scale：设置了页面最大缩放程度  
5、user-scalable：是否允许用户对页面进行缩放操作

下面是一个常用的meta标签实例

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

上面代码的意思是，让布局视口的宽度等于理想视口的宽度，页面的初始缩放比例以及最大缩放比例都为1，且不允许用户对页面进行缩放操作。

媒体查询

媒体查询是响应式设计的基础，他有以下三点作用：

1、检测媒体的类型，比如 screen，tv等  
2、检测布局视口的特性，比如视口的宽高分辨率等  
3、特性相关查询，比如检测浏览器是否支持某某特性（这一点不讨论，因为它被目前浏览器支持的功能对于web开发来讲很无用）

css中使用媒体查询的语法：

@media 媒体类型 and (视口特性阀值){

// 满足条件的css样式代码

}

下面是一段在css中使用媒体查询的示例:

@media all and (min-width: 321px) and (max-width: 400px){

.box{

background: red;

}

}

上面代码中，媒体类型为all，代表任何设备，并且设备的布局视口宽度大于等于321px且小于等于400px时，让拥有box类的元素背景变红。  
这里只是简单介绍了css3的媒体查询，但是用于本系列文章（我也呵呵了，其实就两篇文章这是第一篇，第二篇会尽快更新）已经足够用了，感兴趣或者希望获得更多css3 媒体查询知识的同学可以在网上查阅相关资料。

经过6个小时的奋战，好吧，终于整理好了 《一篇真正教会你开发移动端页面的文章》系列文章的第一篇，这篇文章主要是为[第二篇](http://hcysun.me/2015/10/19/%E4%B8%80%E7%AF%87%E7%9C%9F%E6%AD%A3%E6%95%99%E4%BC%9A%E4%BD%A0%E5%BC%80%E5%8F%91%E7%A7%BB%E5%8A%A8%E7%AB%AF%E9%A1%B5%E9%9D%A2%E7%9A%84%E6%96%87%E7%AB%A0-%E4%BA%8C/)文章要将到的内容打基础，同学们如果不了解可以多阅读即便，如果有问题欢迎评论，我会及时回复的，另外，下一篇文章将会带来干货，真真正正的让大家掌握移动端页面的开发。如果你已经阅读到了这里，我对你表示感谢，你的坚持一定会得到回报，共勉。