

后盾人 人人做后盾

www.houdunren.com

flexible方式的移动端开发

后盾人 2011-2018

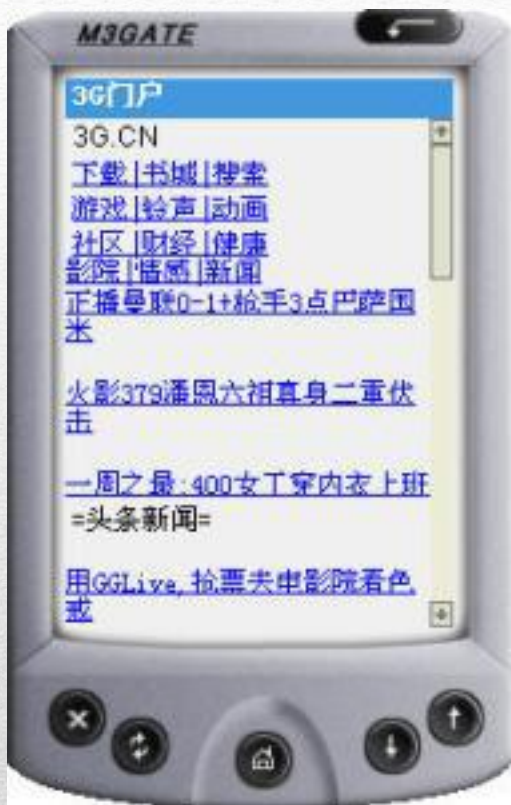
在PC为霸主的时代，任何网页只需要严格按照设计图中标注的具体像素来开发即可，但是一些花哨炫丽的JavaScript特效增大了PC端的开发难度。

岁月的车轮滚到移动端时代，由于移动设备屏幕较小，可利用空间有限，所以没有较多复杂JavaScript特效的需求，但是、可是、但可是.....，不同品牌不同型号的移动设备有着千奇百怪的屏幕，让我们的页面在所有屏幕中都保持良好的展示，这是前端开发工程师面临的一个新问题。

新时代的新问题

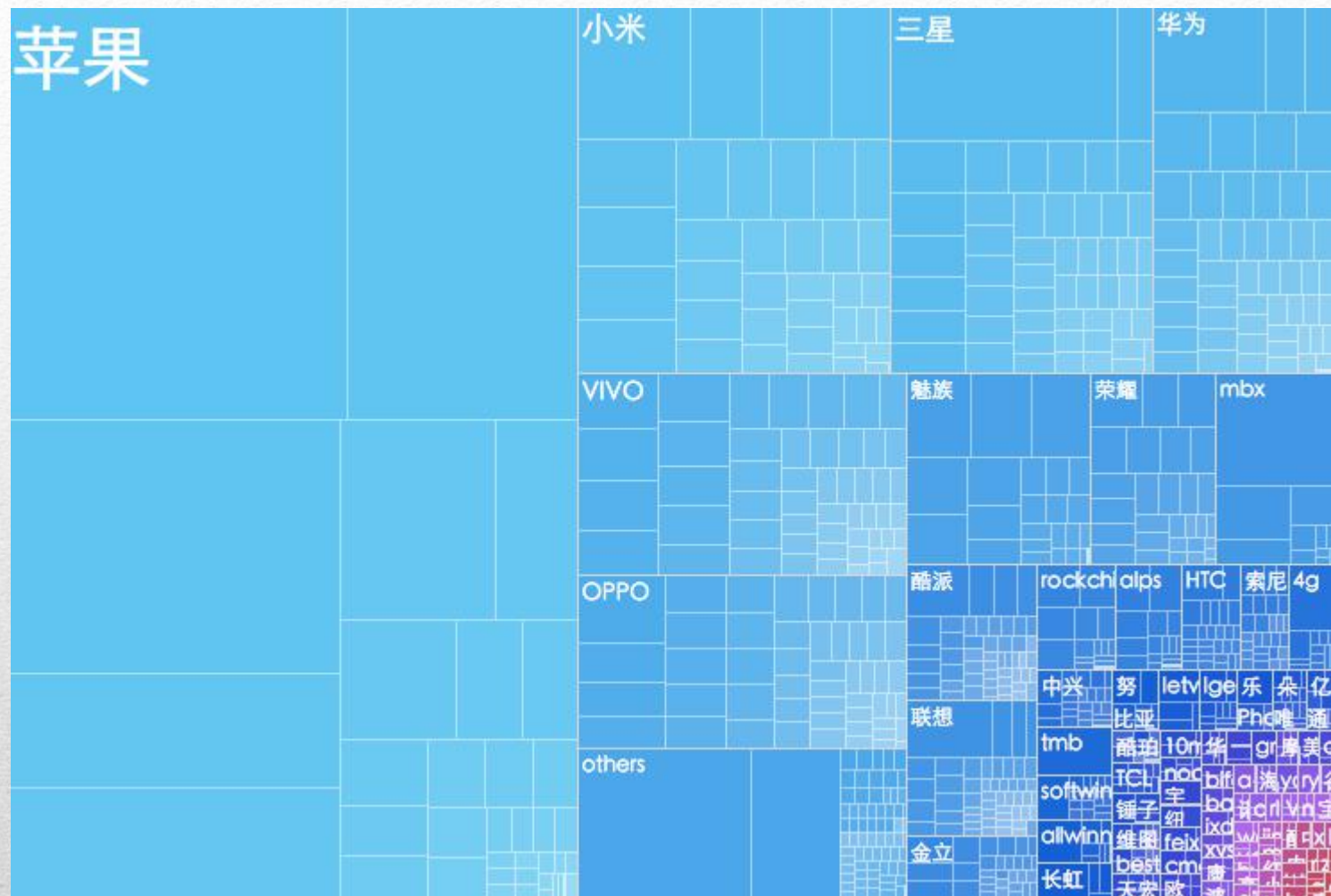
- 1、没有移动端 gprs
- 2、wap页面 gprs时代
- 3、media媒体查询响应式页面 3G时代
- 4、pc端和移动端完全分离 4G
- 5、美好未来 5G

移动端发展的几个阶段
















移动端发展的几个阶段

后盾人 人人做后盾



可怕的杂屏时代

后盾人 人人做后盾

Type	Device	Platform	Screen dimensions in cm	Aspect Ratio	Width × Height dp	Width × Height px	Density
	Asus Zen Watch	Android	1.6 in 1.2 × 1.2 in	1 : 1	213 × 213 dp	320 × 320 px	1.5 hdpi
	Dell Venue 8	Android	8.4 in 4.5 × 7.1 in	16 : 10	800 × 1280 dp	1600 × 2560 px	2.0 xhdpi
	Google Pixel	Android	5.0 in 2.5 × 4.4 in	16 : 9	411 × 731 dp	1080 × 1920 px	2.6 xxhdpi
	Google Pixel XL	Android	5.5 in 2.7 × 4.8 in	16 : 9	411 × 731 dp	1440 × 2560 px	3.5 xxxhdpi
	HTC One M8	Android	5.0 in 2.5 × 4.4 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
	HTC One M9	Android	5.0 in 2.5 × 4.4 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
	LG G Watch	Android	1.7 in 1.2 × 1.2 in	1 : 1	187 × 187 dp	280 × 280 px	1.5 hdpi
	LG G Watch R	Android	1.8 in 1.3 × 1.3 in	1 : 1	213 × 213 dp	320 × 320 px	1.5 hdpi
	LG G2	Android	5.2 in 2.5 × 4.5 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
	LG G3	Android	5.5 in 2.7 × 4.8 in	16 : 9	480 × 853 dp	1440 × 2560 px	3.0 xxhdpi
	Moto 360	Android	1.6 in 1.6 × 1.4 in	32 : 29	241 × 218 dp	320 × 290 px	1.3 tvdpi
	Moto 360 v2 42mm	Android	1.4 in 1.2 × 1.2 in	65 : 64	241 × 244 dp	320 × 325 px	1.3 tvdpi
	Moto 360 v2 46mm	Android	1.6 in 1.4 × 1.4 in	33 : 32	241 × 248 dp	320 × 330 px	1.3 tvdpi

可怕的杂屏时代

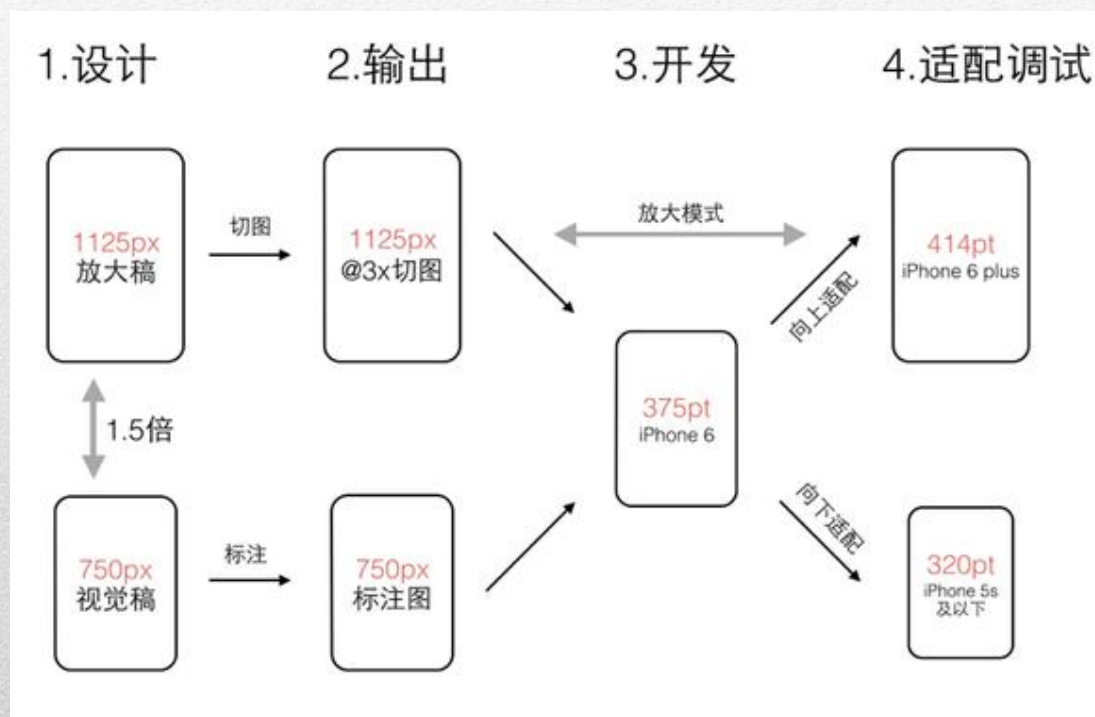
后盾人 人人做后盾

为了应对这多么的终端设备，设计师和前端开发之间又应该采用什么协作模式呢？

- 选择一种尺寸作为设计和开发基准
- 定义一套适配规则，自动适配剩下的多种尺寸
- 特殊适配效果给出设计效果

应对思路

选择iPhone6作为基准设计尺寸，交付给前端的设计尺寸是按750px * 1334px为准(高度会随着内容多少而改变)。前端开发人员通过一套适配规则自动适配到其他的尺寸。



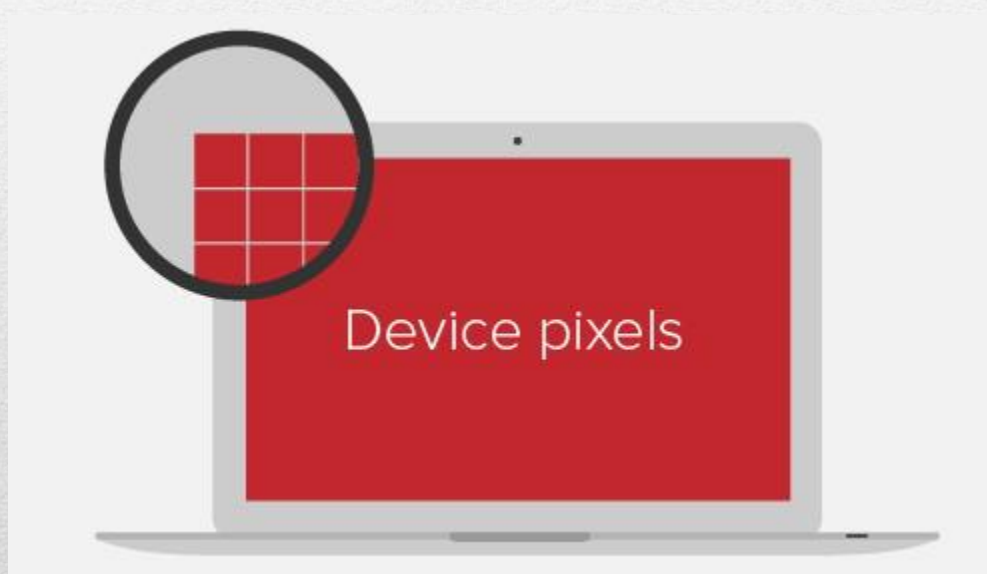
手机淘宝团队适配模式

简单的理解，viewport是严格等于浏览器的窗口。在桌面浏览器中，viewport就是浏览器窗口的宽度高度。但在移动设备上就有点复杂。

*关于viewport的详细解释（对目前知识体系没有太大影响，推荐阅读）：<http://www.w3cplus.com/css/viewports.html>

视窗 viewport

物理像素又被称为设备像素，他是显示设备中一个最微小的物理部件。每个像素可以根据操作系统设置自己的颜色和亮度。正是这些设备像素的微小距离欺骗了我们肉眼看到的图像效果。



物理像素(physical pixel)

简称：dip或dp

设备独立像素也称为密度无关像素，可以认为是计算机坐标系统中的一个点，这个点代表一个可以由程序使用的虚拟像素(比如说CSS像素)，然后由相关系统转换为物理像素。

设备独立像素(density-independent pixel)

后盾人 人人做后盾

CSS像素是一个抽象的单位，主要使用在浏览器上，用来精确度量Web页面上的内容。一般情况之下，CSS像素称为与设备无关的像素(device-independent pixel)，简称DIPs。

CSS像素

屏幕密度是指一个设备表面上存在的像素数量，它通常以每英寸有多少像素来计算(PPI)。

屏幕密度

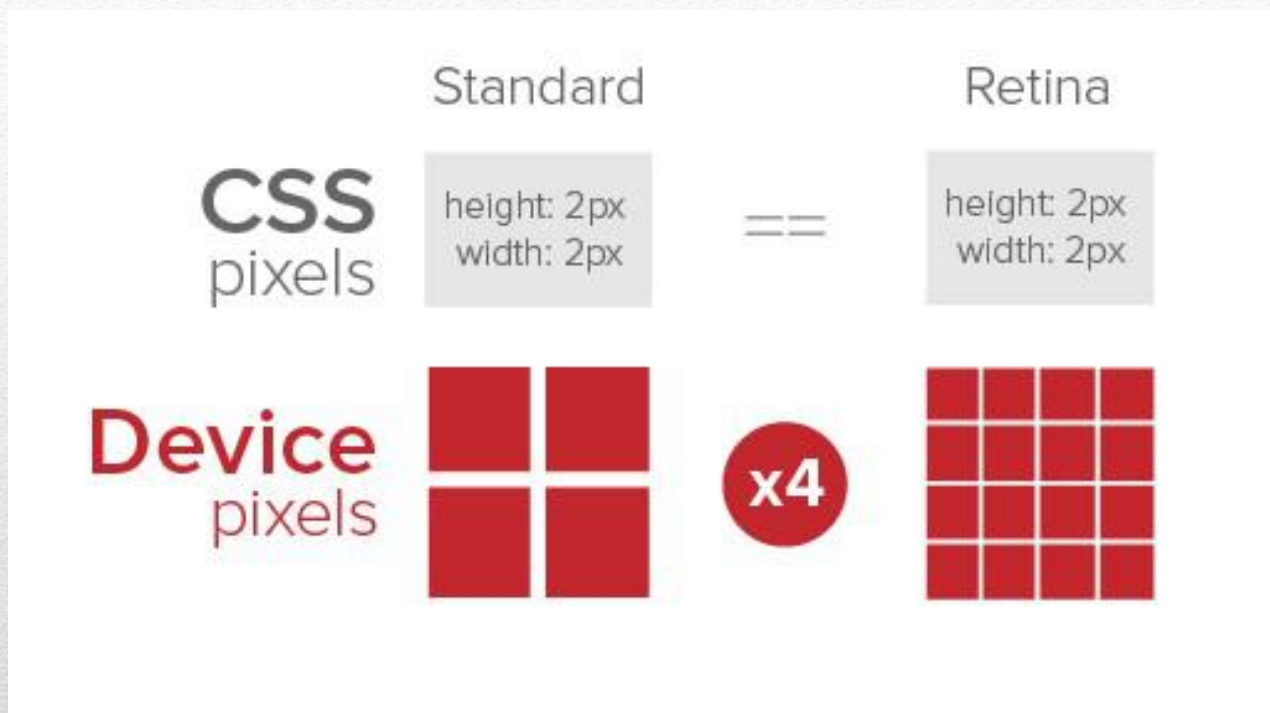
设备像素比简称为dpr，其定义了物理像素和设备独立像素的对应关系。它的值可以按下面的公式计算得到：

$$\text{设备像素比} = \text{物理像素} / \text{设备独立像素}$$

设备像素比(device pixel ratio)

- 在JavaScript中，可以通过window.devicePixelRatio获取到当前设备的dpr。而在CSS中，可以通过-webkit-device-pixel-ratio, -webkit-min-device-pixel-ratio和-webkit-max-device-pixel-ratio进行媒体查询，对不同dpr的设备，做一些样式适配(这里只针对webkit内核的浏览器和webview)。
- dip或dp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。

一些说明



一些说明

- iPhone6的设备宽度和高度为 $375\text{pt} * 667\text{pt}$,可以理解为设备的独立像素
- 而其dpr为2, 根据上面公式, 我们可以很轻松得知其物理像素为 $750\text{pt} * 1334\text{pt}$

分辨率的换算

某元素的CSS样式:

`width: 2px;`

`height: 2px;`

在不同的屏幕上，CSS像素所呈现的物理尺寸是一致的，而不同的是CSS像素所对应的物理像素具数是不一致的。在普通屏幕下1个CSS像素对应1个物理像素，而在Retina屏幕下，1个CSS像素对应的却是4个物理像素。

举例

- <meta> 标签有很多种，而这里要着重说的是viewport的meta 标签，其主要用来告诉浏览器如何规范的渲染Web页面，而你则需要告诉它视窗有多大。在开发移动端页面，我们需要设置 meta 标签如下：

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

- 代码以显示网页的屏幕宽度定义了视窗宽度。网页的比例和最大比例被设置为100%。
- 我们的解决方案中重度依赖meta 标签

meta 标签

简单的理解，rem就是相对于根元素<html>的font-size来做计算。而我们的方案中使用rem单位，是能轻易的根据<html>的font-size计算出元素的盒模型大小。而这个特色对我们来说是特别的有益处。

rem

- Flexible会将视觉稿分成100份（主要为了以后能更好的兼容vh和vw），而每一份被称为一个单位a。同时1rem单位被认定为10a。
- 对于视觉稿上的元素尺寸换算，只需要原始的px值除以rem基准值即可。假如设计稿宽度为750px,其中某div宽度为200px*200px，尺寸可以转换成为：
- $200/75 * 200/75 = 2.666\text{rem} * 2.666\text{rem}$

flexible原理

1. 直接用阿里提供的官方cdn:

```
<script src="http://g.tbcdn.cn/mtb/lib-flexible/0.3.4/??flexible_css.js,flexible.js"></script>
```

2. 将插件下载到本地，然后引入flexible_css.js和flexible.js文件

flexible的引入

1. 元素宽高一律用rem来指定
2. 不必手动指定根元素的font-size
3. 不必手动指定meta标签

开发要点

4. 文字大小用px, 如:

```
div {  
    width: 1rem;  
    height: 0.4rem;  
    font-size: 12px; // 默认写上dpr为1的font-size  
}  
  
[data-dpr="2"] div { font-size: 24px; }  
[data-dpr="3"] div { font-size: 36px; }
```

开发要点
