

H5 移动设配适配方案

一、移动端一些概念

◆ 视觉稿 (选取一款手机的屏幕宽高作为**基准**)

在前端开发之前，视觉 MM 会给我们一个 **psd 文件**，称之为**视觉稿**。

对于移动端开发而言，为了做到页面高清的效果，视觉稿的规范往往会遵循以下两点：

1) 首先，选取一款手机的屏幕宽高作为**基准**(以前是 iPhone4 的 320×480 ，现在更多的是 iPhone6 的 375×667)。

2) 对于 retina 屏幕(如: $dpr=2$)，为了达到高清效果，视觉稿的**画布大小**会是基准的 2 倍，也就是说像素点个数是原来的 4 倍 (对 iPhone6 而言：原先的 375×667 ，就会变成 750×1334)。

问题：

对于 $dpr=2$ 的手机，为什么画布大小 $\times 2$ ，就可以解决高清问题？

对于 2 倍大小的视觉稿，在具体的 css 编码中如何还原每一个区块的真实宽高(也就是布局问题)？

◆ 标注稿

物理像素(physical pixel)

一个物理像素是显示器(手机屏幕)上最小的物理显示单元，在操作系统的调度下，每一个设备像素都有自己的颜色值和亮度值。

设备独立像素(density-independent pixel)

设备独立像素(也叫密度无关像素)，可以认为是计算机坐标系统中得一个点，这个点代表一个可以由程序使用的虚拟像素(比如: css 像素)，然后由相关系统转换为物理像素。

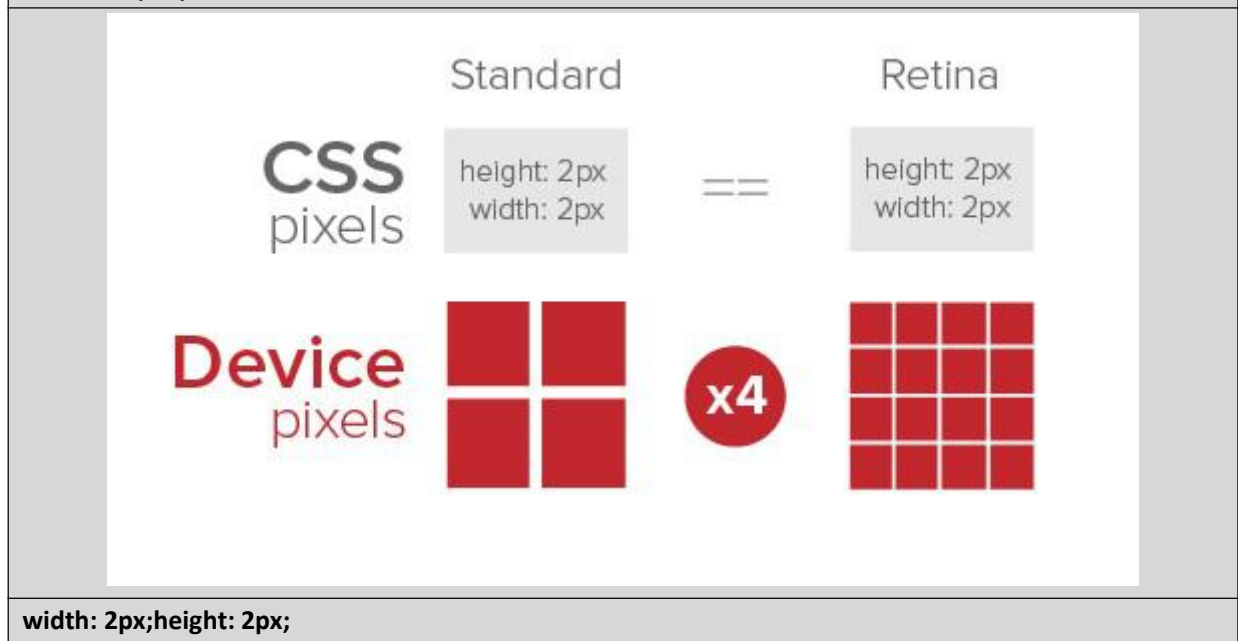
设备像素比(device pixel ratio)

设备像素比(简称 dpr)定义了物理像素和设备独立像素的对应关系，它的值可以按如下的公式的得到： 设备像素比 = 物理像素 / 设备独立像素 // 在某一方向上，x 方向或者 y 方向。

在 Javascript 中，可以通过 `window.devicePixelRatio` 获取到当前设备的 dpr。

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)。

在普通屏幕下，1 个 css 像素 对应 1 个物理像素(1:1)。 在 retina 屏幕下，1 个 css 像素对应 4 个物理像素(1:4)。



◆ 位图像素

一个位图像素是栅格图像(如: png, jpg, gif 等)最小的数据单元。每一个位图像素都包含着一些自身的显示信息(如: 显示位置, 颜色值, 透明度等)。

retina 下图片的展示情况?

理论上, 1 个位图像素对应于 1 个物理像素, 图片才能得到完美清晰的展示。
在普通屏幕下是没有问题的, 但是在 retina 屏幕下就会出现位图像素点不够, 从而导致图片模糊的情况。



如上图: 对于 dpr=2 的 retina 屏幕而言, 1 个位图像素对应于 4 个物理像素, 由于单个位图像素不可以再进一步分割, 所以只能就近取色, 从而导致图片模糊(注意上述的几个颜色值)。

所以, 对于图片高清问题, 比较好的方案就是两倍图片(@2x)。如: 200×300(css pixel)img 标签, 就需要提供 400×600 的图片。

如此一来, 位图像素点个数就是原来的 4 倍, 在 retina 屏幕下, 位图像素点个数就可以跟物理像素点个数形成 1:1 的比例, 图片自然就清晰了(这也解释了之前留下的一个问题, 为啥视觉稿的画布大小要×2?)。

这里就还有另一个问题, 如果普通屏幕下, 也用了两倍图片, 会怎样呢?
很明显, 在普通屏幕下, 200×300(css pixel)img 标签, 所对应的物理像素个数就是 200×300 个, 而两倍图片的位图像素个数则是 200×300×4, 所以就出现一个物理像素点对应 4 个位图像素点,

所以它的取色也只能通过一定的**算法**(显示结果就是一张只有原图像素总数四分之一, 我们称这个过程叫做 **downsampling**), 肉眼看上去虽然图片不会模糊, 但是会觉得图片缺少一些锐利度, 或者是有点色差(但还是可以接受的)。



◆ Retina 下, 图片高清问题

所以最好的解决办法是：不同的 dpr 下，加载不同的尺寸的图片。

不管是通过 css 媒体查询，还是通过 javascript 条件判断都是可以的。

那么问题来了，这样的话，不就是要准备两套图片了嘛？(@1x 和@2x)

我想，做的好的公司，都会有这么一个**图片服务器**，通过 url 获取参数，然后可以控制图片质量，也可以将图片裁剪成不同的尺寸。

所以我们只需上传大图(@2x)，其余小图都交给图片服务器处理，我们只要负责拼接 url 即可。

◆ Retina 下，border: 1px 问题

设计师想要的 retina 下 border: 1px;，其实就是 1 物理像素宽，对于 css 而言，可以认为是 border: 0.5px;，这是 retina 下(dpr=2)下能显示的最小单位。

然而，无奈并不是所有手机浏览器都能识别 border: 0.5px;，ios7 以下，android 等其他系统里，0.5px 会被当成为 0px 处理，那么如何实现这 0.5px 呢？

方案一： 最简单的一个做法就是这样(元素 scale):

```
.scale{ position: relative;}
```

```
.scale:after {content: ""; position: absolute; bottom: 0px; left: 0px; right: 0px; border-bottom: 1px solid #ddd; -webkit-transform: scaleY(.5); -webkit-transform-origin: 0 0;}
```

方案一问题：

通过 transform: scaleY(.5)缩小 0.5 倍来达到 0.5px 的效果，但是这样 hack 实在是不够通用(如：圆角等)。

方案二： 页面 scale 的方案，是比较通用的，几乎满足所有场景。

对于 iphone5(dpr=2)，添加如下的 meta 标签，设置 viewport(scale 0.5):

页面 scale，必然会带来一些问题：

- 1) 字体大小会被缩放
- 2) 页面布局会被缩放(如: div 的宽高等)

◆ 多屏适配布局问题

移动端布局，为了适配各种大屏手机，目前最好用的方案莫过于使用**相对单位 rem（等比）**。
基于 rem 的原理，我们要做的就是：
针对不同手机屏幕尺寸和 dpr 动态的改变根节点 html 的 font-size 大小(基准值)。

这里我们提取了一个公式(rem 表示基准值)

```
rem = document.documentElement.clientWidth * dpr / 10
```

说明：

- 1) 乘以 dpr，是因为页面有可能为了实现 1px border 页面会缩放(scale) 1/dpr 倍(如果没有，dpr=1)。
- 2) 除以 10，是为了取整，方便计算(理论上可以是任何值)

所以就像下面这样，html 的 font-size 可能会：

iPhone3gs: 320px / 10 = 32px

iPhone4/5: 320px * 2 / 10 = 64px

iPhone6: 375px * 2 / 10 = 75px

对于动态改变根节点 html 的 font-size，我们可以通过 css 做，也可以通过 javascript 做。

如果有一个区块，在 psd 文件中量出：宽高 750×300px 的 div，那么如何转换成 rem 单位呢？

换算 Rem 公式如下：rem = px / 基准值;

对于一个 iPhone6 的视觉稿，它的基准值就是 75(之前有提到);

二、移动端适配方案(flexible 方案)

◆ lib-flexible 说明

在整个手淘团队，我们有一个名叫 lib-flexible 的库，而这个库就是用来解决 H5 页面终端适配的。
lib-flexible 是一个制作 H5 适配的开源库，

下载：<https://github.com/amfe/lib-flexible/archive/master.zip>

◆ 使用方法

lib-flexible 库的使用方法非常的简单，只需要在 Web 页面的<head></head>中添加对应的 flexible_css.js,flexible.js 文件：

```
<script src="build/flexible_css.js"></script>
<script src="build/flexible.js"></script>
```

事实上他做了这几样事情：

1. 动态改写<meta>标签
2. 给<html>元素添加 data-dpr 属性，并且动态改写 data-dpr 的值

3. 给<html>元素添加 font-size 属性，并且动态改写 font-size 的值

快速 px 转为 rem 工具

CSSREM 是一个 CSS 的 px 值转 rem 值的 Sublime Text3 自动完成插件。这个插件是由@正霖编写。

下载地址：<https://github.com/flashlizi/cssrem>

安装

进入 packages 目录：Sublime Text -> Preferences -> Browse Packages...

复制下载的 cssrem 目录到刚才的 packages 目录里。

重启 Sublime Text。

配置参数

参数配置文件：Sublime Text -> Preferences -> Package Settings -> cssrem

px_to_rem - px 转 rem 的单位比例，默认为 40。

max_rem_fraction_length - px 转 rem 的小数部分的最大长度。默认为 6。

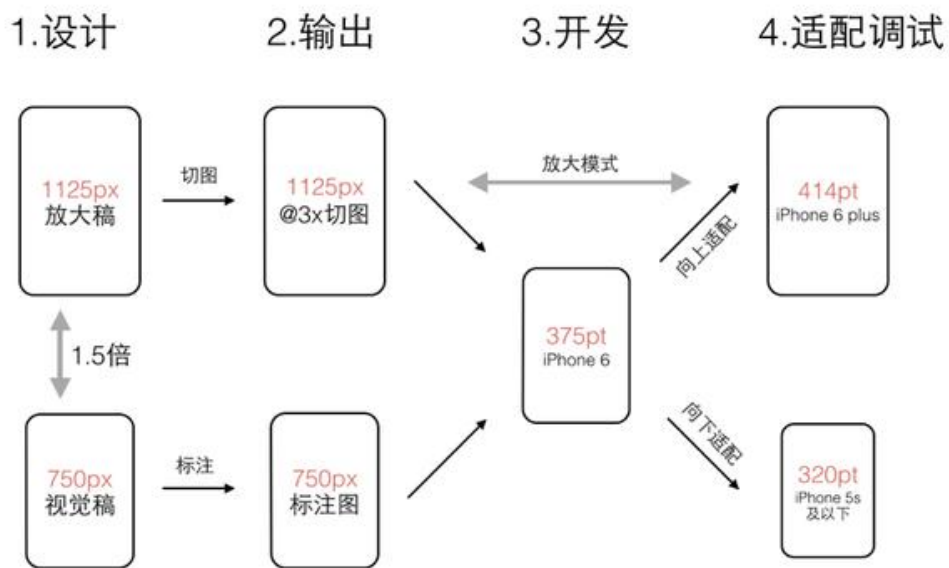
available_file_types - 启用此插件的文件类型。默认为：[".css", ".less", ".sass"]。

字号不使用 rem

我们在 iPhone3G 和 iPhone4 的 Retina 屏下面，希望看到的文本字号是相同的。也就是说，我们不希望文本在 Retina 屏幕下变小，另外，我们希望在大屏手机上看到更多文本，以及，现在绝大多数的字体文件都自带一些点阵尺寸，通常是 16px 和 24px，所以我们不希望出现 13px 和 15px 这样的奇葩尺寸。

```
.selector {
  width: 2rem;
  border: 1px solid #ddd;
}
[data-dpr="1"] .selector {
  height: 32px;
  font-size: 14px;
}
[data-dpr="2"] .selector {
  height: 64px;
  font-size: 28px;
}
[data-dpr="3"] .selector {
  height: 96px;
  font-size: 42px;
}
```

手机淘宝团队适配协作模式



手淘设计师常选择 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。

参考网址:

<http://web.jobbole.com/84285/>

<https://github.com/amfe/lib-flexible>

<http://www.w3cplus.com/css/viewports.html>