***主要研究目标：***

1. ***宽高比：一般是2倍、3倍***
2. ***X2、X3 高清比***
3. ***图片展现方式，img标签 …..***

***4、手机端的约束视口一般控制在320px-480px***

**前言：**

移动端设备现在有平板和手机，简单来说，移动端页面就是只适应了平板和手机的视口。这里又涉及到了什么是视口，视口只是我个人平时习惯而已，其实就是指设备的可视窗的宽度。

说起可视窗，web前端的朋友们相当熟悉了。下面我依然用视口来代替可视窗。视口，有默认视口和约束视口，默认视口是980px，约束视口是手机厂商自己规定的，所有现在市面上的安卓机和苹果机的约束视口大小都在320px-480px之间。所以只要你的网页在320-480之间没有问题，页面接口不崩塌，那就是一个没问题的移动页面。

响应式适应各种设备，而移动端就是为移动设备服务的。那有人就说了，那不如都写响应式得了，还写什么移动端页面，没错，话是这么说没错，但是你会发现，之所以一个网页能做成响应式，他的布局都是经过精心设计的，你像淘宝京东就做不了响应式，他怎么响，随便页面一缩布局留塌，所以对于这种内容较多的电商网站，得分开做，一套pc，一套移动端。

*/\*  
\* 设计稿一般是根据iPhone6手机进行页面设计的；  
\* iPhone6的理想视口（viewport）是375px  
\* iPhone6的设备像素比为2，所以设备像素为750px;  
\* 所以，用单位REM。REM会根据HTML元素的font-size进行设置，当HTML元素的font-size： 16px时1rem=16px;  
\* 基本思路：  
\* 通过window.screen.width获取不同移动设备的理想视口宽度。  
\* 规定基准值为750px(此值为iPhon6的设备像素)。  
\* (1) / (2) \* 100即得到HTML元素的font-size。(乘于100是因为我们在前面将字体缩小了100倍，此时要乘回来)  
\* 换算成公式即：设计稿尺寸 / 100 \* (不同设备的理想视口宽度 / 基准值 \* 100)  
\* \*/***var *ideaViewWidth*** = window.screen.**width**;  
**var *basevalue*** = 750;  
document.documentElement.**style**.**fontSize** = (***ideaViewWidth***/***basevalue***) \* 100 + **'px'**;  
  
console.log(***ideaViewWidth***/***basevalue***\*100)

**响应式布局：**

1、百分比布局

2、弹性图片

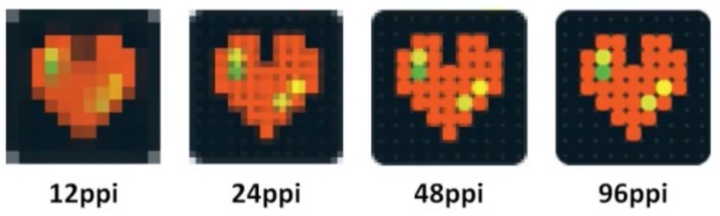
Img{ max-width: 100%;height:auto;}

3、重新布局，显示与隐藏。

显示比重高的模块使用绝对定位

特别小设备下，隐藏一些内容或者模块

**高清图片：**

如上图

同一张图片，在普通屏显示正常，但高清屏出现模糊。原因是原来一个像素的点分成的四个像素的点进行了显示。

解决方案：在高清屏中把图片变成二倍图,前提是二倍的高清图已经存在。

.icon{

background: url(bg.png) no-repeat; /\* 宽200px; \*/

}

@media screen and (-webkit-min-device-pixel-ratio:1.5){

.icon{

background-image: url(bgx2.png); /\* 宽400px; \*/

background-size: 200px; /\* 宽200px; \*/

}

}

**一像素边框：**

**1、通过transform: scale(0.5)**



2、图片方案

**相对单位REM：**

为了适应各大屏幕的手机，px略显固定，不能根据尺寸的大小而改变，用相对单位更能体验页面的兼容性。

em: 是根据父节点的font-size为相对单位

rem: 是根据html的font-size为相对单位

em 多重嵌套下，变得非常难以为何，

rem更加能作为全局统一设置的度量

不用rem的情况：font-size

**多行文本溢出：**

*单行文本溢出*

<div style = 'width:400px;

height:40px;

border:1px solid red;'>

<p style='overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;'>

这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本

</p>

</div>

*多行文本溢出*

<div style = 'width:400px;

height:70px;

border:1px solid red;'>

<p style='display:

-webkit-box;//对象作为弹性伸缩盒子模型显示

-webkit-box-orient: vertical;//设置或检索伸缩盒对象的子元素的排列方式

-webkit-line-clamp: 2;//溢出省略的界限

overflow:hidden;//设置隐藏溢出元素'>

这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本

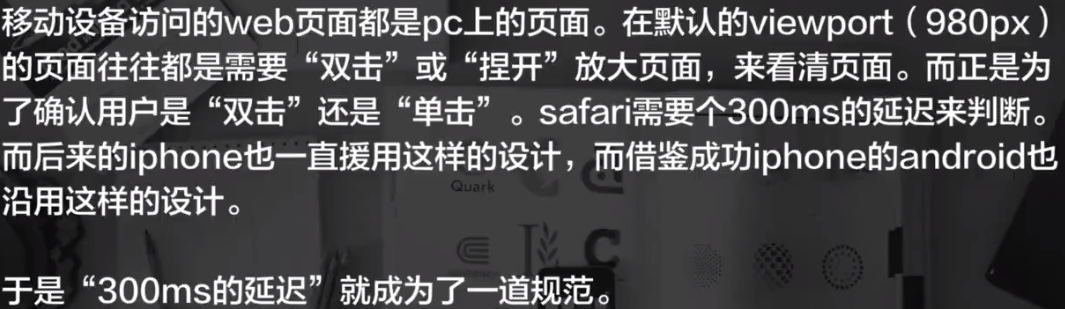
</p>

</div>

**移动终端交互优化：**

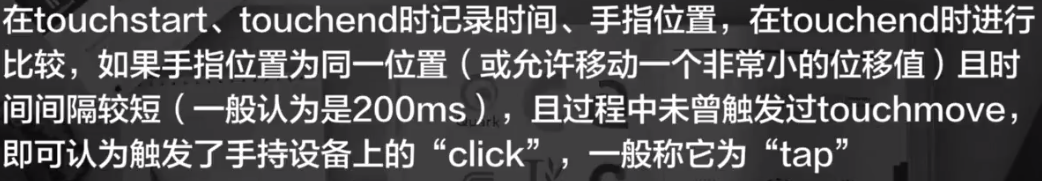
**300毫秒的故事**

移动web页面上的click事件响应都要慢上300Ms



300ms延迟破解：使用Tap事件答题click事件

自定义Tap事件的原理：



**Touch基础事件：**