



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

移动WEB开发之流式布局

目录 Contents

- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端调试
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局
- ◆ 移动端开发之流式布局
- ◆ 京东移动端首页制作

1. 移动端基础

1.1 浏览器现状

PC端常见浏览器

360浏览器、谷歌浏览器、火狐浏览器、QQ浏览器、百度浏览器、搜狗浏览器、IE浏览器。



移动端常见浏览器

UC浏览器，QQ浏览器，欧朋浏览器，百度手机浏览器，360安全浏览器，谷歌浏览器，搜狗手机浏览器，猎豹浏览器，以及其他杂牌浏览器。

国内的UC和QQ，百度等手机浏览器都是根据Webkit修改过来的内核，国内尚无自主研发的内核，就像国内的手机操作系统都是基于Android修改开发的一样。

总结：兼容移动端主流浏览器，处理Webkit内核浏览器即可。

1. 移动端基础

1.2 手机屏幕现状

- 移动端设备屏幕尺寸非常多，碎片化严重。
- Android设备有多种分辨率：480x800, 480x854, 540x960, 720x1280, 1080x1920等，还有传说中的2K，4k屏。
- 近年来iPhone的碎片化也加剧了，其设备的主要分辨率有：640x960, 640x1136, 750x1334, 1242x2208等。
- 作为开发者无需关注这些分辨率，因为我们常用的尺寸单位是 px 。



1. 移动端基础

1.3 常见移动端屏幕尺寸

设备	尺寸(英寸)	开发尺寸(px)	物理像素比(dpr)
iphone3G	3.5	320*480	1.0
iphone4/4s	3.5	320*480	2.0
iphone5/5s/5c	4.0	320*568	2.0
HTC One M8	4.5	360*640	3.0
iphone6	4.7	375*667	2.0
Nexus 4	4.7	384*640	2.0
Nexus 5x	5.2	411*731	2.6
iphone6 Plus	5.5	414*736	3.0
Samsung Galaxy Note 4	5.7	480*853	3.0
Sony Xperia Z Ultra	6.4	540*960	2.0
Nexus 7 ('12)	7.0	600*960	1.3
iPad Mini	7.9	768*1024	1.0

注：以上数据均参考自

<https://material.io/devices/>。

注：作为前端开发，不建议大家去
纠结dp, dpi, pt, ppi等单位。

1. 移动端基础

1.4 移动端调试方法

- Chrome DevTools（谷歌浏览器）的模拟手机调试
- 搭建本地web服务器，手机和服务器一个局域网内，通过手机访问服务器
- 使用外网服务器，直接IP或域名访问

1. 移动端基础

1.4 总结

- 移动端浏览器我们主要对webkit内核进行兼容
- 我们现在开发的移动端主要针对手机端开发
- 现在移动端碎片化比较严重，分辨率和屏幕尺寸大小不一
- 学会用谷歌浏览器模拟手机界面以及调试

目录

Contents

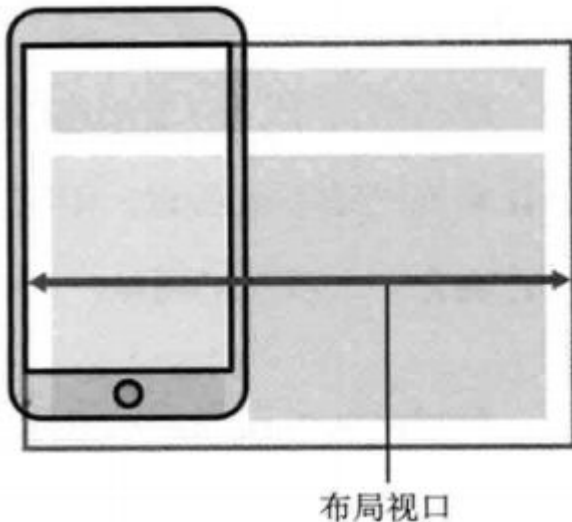
- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端调试
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局

2. 视口

视口 (viewport) 就是浏览器显示页面内容的屏幕区域。视口可以分为布局视口、视觉视口和理想视口

2.1 布局视口 layout viewport

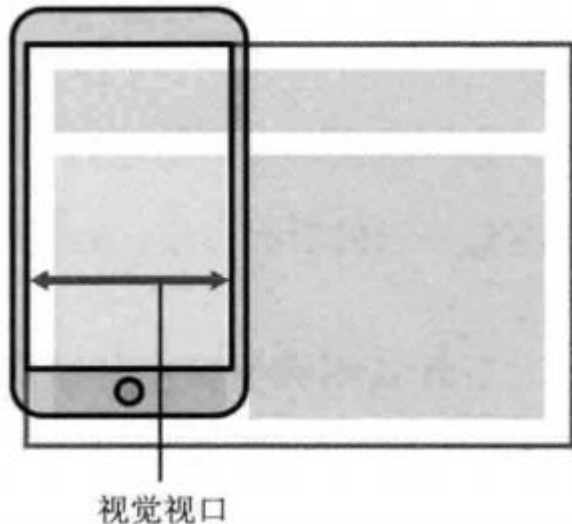
- 一般移动设备的浏览器都默认设置了一个布局视口，用于解决早期的PC端页面在手机上显示的问题。
- iOS, Android基本都将这个视口分辨率设置为 980px，所以PC上的网页大多都能能在手机上呈现，只不过元素看上去很小，一般默认可以通过手动缩放网页。



2. 视口

2.2 视觉视口 visual viewport

- 字面意思，它是用户正在看到的网站的区域。**注意：是网站的区域。**
- 我们可以通过缩放去操作视觉视口，但不会影响布局视口，布局视口仍保持原来的宽度。



2. 视口

2.3 理想视口 **ideal viewport**

- 为了使网站在移动端有最理想的浏览和阅读宽度而设定
- 理想视口，对设备来讲，是最理想的视口尺寸
- 需要手动添写meta视口标签通知浏览器操作
- meta视口标签的主要目的：布局视口的宽度应该与理想视口的宽度一致，简单理解就是设备有多宽，我们布局的视口就多宽

2. 视口

2.4 总结

- 视口就是浏览器显示页面内容的屏幕区域
- 视口分为布局视口、视觉视口和理想视口
- 我们移动端布局想要的是理想视口就是手机屏幕有多宽，我们的布局视口就有多宽
- 想要理想视口，我们需要给我们的移动端页面添加 meta视口标签

2.5 meta视口标签

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

属性	解释说明
width	宽度设置的是viewport宽度，可以设置device-width特殊值
initial-scale	初始缩放比，大于0的数字
maximum-scale	最大缩放比，大于0的数字
minimum-scale	最小缩放比，大于0的数字
user-scalable	用户是否可以缩放，yes或no（1或0）

2. 视口

2.6 标准的viewport设置

- 视口宽度和设备保持一致
- 视口的默认缩放比例1.0
- 不允许用户自行缩放
- 最大允许的缩放比例1.0
- 最小允许的缩放比例1.0

目录 Contents

- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端简介
- ◆ 移动端常见布局
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局

3.1 物理像素&物理像素比

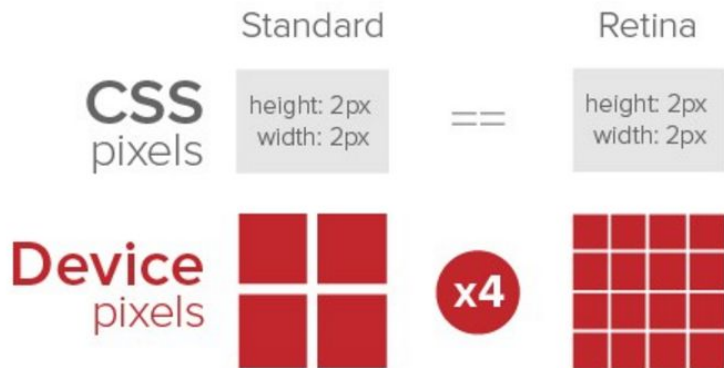
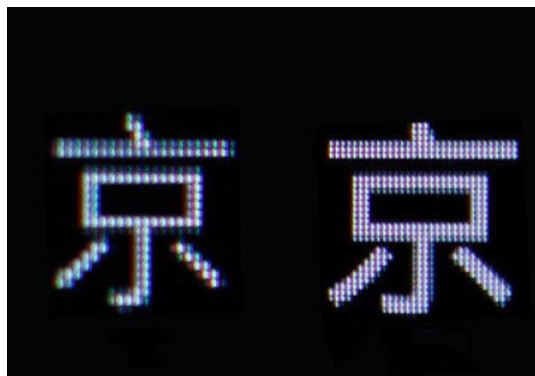
- 物理像素点指的是屏幕显示的最小颗粒，是物理真实存在的。这是厂商在出厂时就设置好了,比如苹果6\7\8 是 750* 1334
- 我们开发时候的1px 不一定等于1个物理像素的
- PC端页面，1个px 等于1个物理像素的，但是移动端就不尽相同
- 一个px的能显示的物理像素点的个数，称为物理像素比或屏幕像素比

设备	尺寸(英寸)	开发尺寸(px)	物理像素比(dpr)
iphone3G	3.5	320*480	1.0
iphone4/4s	3.5	320*480	2.0
iphone5/5s/5c	4.0	320*568	2.0
HTC One M8	4.5	360*640	3.0
iphone6	4.7	375*667	2.0
Nexus 4	4.7	384*640	2.0
Nexus 5x	5.2	411*731	2.6
iphone6 Plus	5.5	414*736	3.0
Samsung Galaxy Note 4	5.7	480*853	3.0
Sony Xperia Z Ultra	6.4	540*960	2.0
Nexus 7 ('12)	7.0	600*960	1.3
iPad Mini	7.9	768*1024	1.0

3. 二倍图

3.1 物理像素&物理像素比

- PC端 和 早前的手机屏幕 / 普通手机屏幕: 1CSS像素 = 1 物理像素的
- Retina (视网膜屏幕) 是一种显示技术, 可以将把更多的物理像素点压缩至一块屏幕里, 从而达到更高的分辨率, 并提高屏幕显示的细腻程度。



3.2 多倍图

- 对于一张 50px * 50px 的图片,在手机 Retina 屏中打开,按照刚才的物理像素比会放大倍数,这样会造成图片模糊
- 在标准的viewport设置中,使用倍图来提高图片质量,解决在高清设备中的模糊问题
- 通常使用二倍图,因为iPhone 6\7\8 的影响,但是现在还存在3倍图4倍图的情况,这个看实际开发公司需求
- 背景图片 注意缩放问题

```
/* 在 iphone8 下面 */  
img{  
    /*原始图片100*100px*/  
    width: 50px;  
    height: 50px;  
}  
.box{  
    /*原始图片100*100px*/  
    background-size: 50px 50px;  
}
```

3.3 背景缩放 background-size

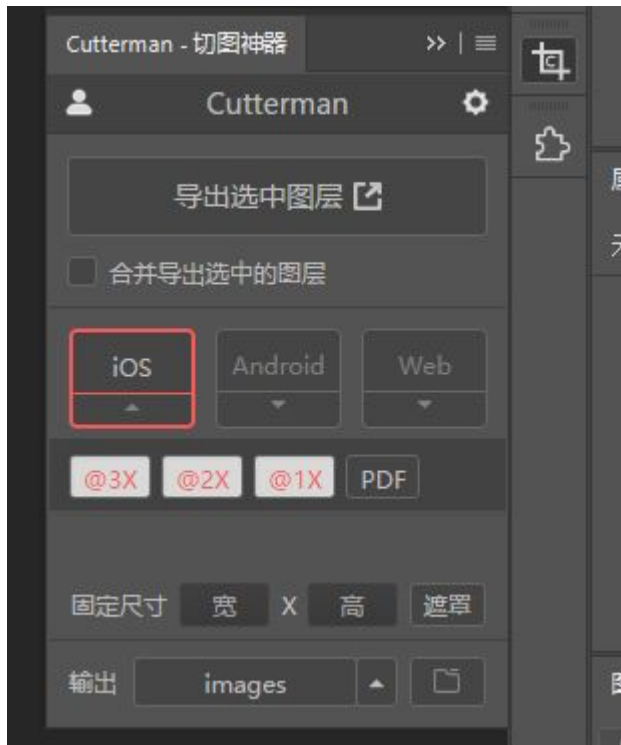
background-size 属性规定背景图像的尺寸

background-size: 背景图片宽度 背景图片高度;

- 单位: 长度 | 百分比 | cover | contain;
- cover把背景图像扩展至足够大, 以使背景图像完全覆盖背景区域。
- contain把图像扩展至最大尺寸, 以使其宽度和高度完全适应内容区域

3. 二倍图

3.4 多倍图切图 cutterman



- @3X 3倍图
- @2X 2倍图
- @1X 1倍图原图

目录 Contents

- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端开发选择
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局

4. 移动端开发选择

4.1 移动端主流方案

1. 单独制作**移动端**页面（主流）

京东商城手机版

淘宝触屏版

苏宁易购手机版

....

2. **响应式**页面兼容移动端（其次）

三星手机官网

....

4. 移动端开发选择

4.2 单独移动端页面（主流）

通常情况下，网址域名前面加 **m(mobile)** 可以打开移动端。通过判断设备，如果是移动设备打开，则跳到**移动端页面**。

m.taobao.com



m.jd.com



m.suning.com



4. 移动端开发选择

4.3 响应式兼容PC移动端

三星电子官网：www.samsung.com/cn/，通过判断屏幕宽度来改变样式，以适应不同终端。

缺点：制作麻烦，需要花很大精力去调兼容性问题



4. 移动端开发选择

4.4 总结

现在市场常见的移动端开发有 单独制作移动端页面 和 响应式页面 两种方案

现在市场主流的选择还是单独制作移动端页面

目录 Contents

- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端开发选择
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局

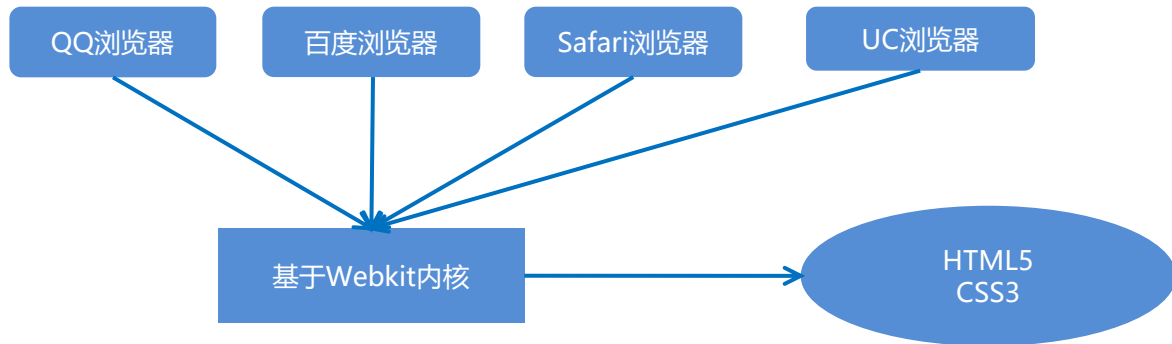
5. 移动端技术解决方案

5.1 移动端浏览器

移动端浏览器基本以 webkit 内核为主，因此我们就考虑webkit兼容性问题。

我们可以放心使用 H5 标签和 CSS3 样式。

同时我们浏览器的私有前缀我们只需要考虑添加 webkit 即可



5. 移动端技术解决方案

5.2 CSS初始化 **normalize.css**

移动端 CSS 初始化推荐使用 normalize.css/

- Normalize.css: 保护了有价值的默认值
- Normalize.css: 修复了浏览器的bug
- Normalize.css: 是模块化的
- Normalize.css: 拥有详细的文档

官网地址: <http://necolas.github.io/normalize.css/>

5. 移动端技术解决方案

5.3 CSS3 盒子模型 **box-sizing**

- 传统模式宽度计算: 盒子的宽度 = CSS中设置的width + border + padding
- CSS3盒子模型: 盒子的宽度 = CSS中设置的宽度width 里面包含了 border 和 padding
也就是说, 我们的CSS3中的盒子模型, padding 和 border 不会撑大盒子了

```
/*CSS3盒子模型*/  
box-sizing: border-box;  
/*传统盒子模型*/  
box-sizing: content-box;
```

传统or CSS3盒子模型?

- 移动端可以全部CSS3 盒子模型
- PC端如果完全需要兼容, 我们就用传统模式, 如果不考虑兼容性, 我们就选择 CSS3 盒子模型

5. 移动端技术解决方案

5.4 特殊样式

```
/*CSS3盒子模型*/  
box-sizing: border-box;  
-webkit-box-sizing: border-box;  
/*点击高亮我们需要清除清除  设置为transparent 完成透明*/  
-webkit-tap-highlight-color: transparent;  
/*在移动端浏览器默认的外观在ios上加上这个属性才能给按钮和输入框自定义样式*/  
-webkit-appearance: none;  
/*禁用长按页面时的弹出菜单*/  
img,a { -webkit-touch-callout: none; }
```

目录 Contents

- ◆ 移动端基础
- ◆ 视口
- ◆ 二倍图
- ◆ 移动端开发选择
- ◆ 移动端技术解决方案
- ◆ 移动端常见布局

6.移动端常见布局

移动端技术选型

移动端布局和以前我们学习的PC端有所区别：

1. 单独制作**移动端**页面（主流）

- 流式布局（百分比布局）
- flex 弹性布局（强烈推荐）
- less+rem+媒体查询布局
- 混合布局

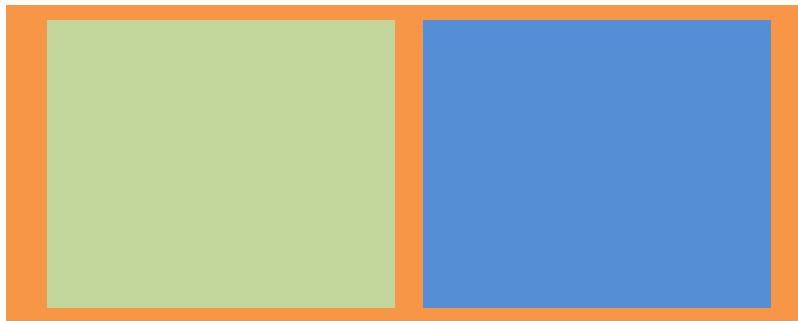
2. **响应式**页面兼容移动端（其次）

- 媒体查询
- bootstrap

6. 移动端常见布局

6.1 流式布局（百分比布局）

- 流式布局，就是百分比布局，也称非固定像素布局。
- 通过盒子的宽度设置成百分比来根据屏幕的宽度来进行伸缩，不受固定像素的限制，内容向两侧填充。
- 流式布局方式是移动web开发使用的比较常见的布局方式。



- max-width 最大宽度 (max-height 最大高度)
- min-width 最小宽度 (min-height 最小高度)

6. 移动端常见布局



案例：京东移动端首页

访问地址：m.jd.com



6. 移动端常见布局



案例：京东移动端首页

1. 技术选型

方案：我们采取单独制作移动页面方案

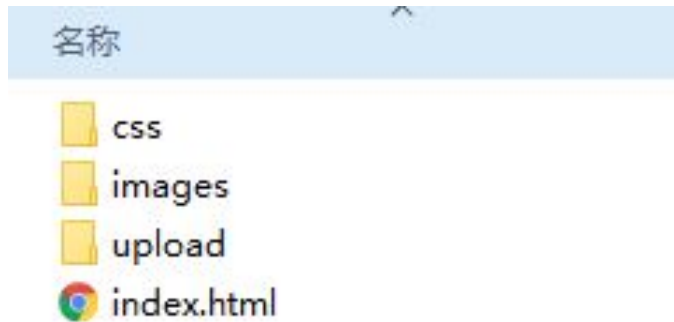
技术：布局采取流式布局

6. 移动端常见布局



案例：京东移动端首页

2. 搭建相关文件夹结构



6. 移动端常见布局



案例：京东移动端首页

3. 设置视口标签以及引入初始化样式

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

```
<link rel="stylesheet" href="css/normalize.css">  
<link rel="stylesheet" href="css/index.css">
```

6. 移动端常见布局



案例：京东移动端首页

4. 常用初始化样式

```
body {  
    margin: 0 auto;  
    min-width: 320px;  
    max-width: 640px;  
    background: #fff;  
    font-size: 14px;  
    font-family: -apple-system, Helvetica, sans-serif;  
    line-height: 1.5;  
    color: #666;  
}
```

6. 移动端常见布局



案例：京东移动端首页

5. 二倍精灵图做法

- 在firework里面把精灵图等比例缩放为原来的一半
- 之后根据大小 测量坐标
- 注意代码里面background-size也要写： 精灵图原来宽度的一半

6. 移动端常见布局



案例：京东移动端首页

6. 图片格式

DPG图片压缩技术

京东自主研发推出DPG图片压缩技术，经测试该技术，可直接节省用户近50%的浏览流量，极大的提升了用户的网页打开速度。

能够兼容jpeg，实现全平台、全部浏览器的兼容支持，经过内部和外部上万张图片的人眼浏览测试后发现，压缩后的图片和webp的清晰度对比没有差距。

webp 图片格式

谷歌开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器宽带资源和数据空间

总结

移动端布局之流式布局

1. 标准viewport规范以及写法
2. 模拟移动端调试方法
3. 移动端常见的布局方案
4. 流式布局原理
5. 京东移动端首页布局技巧



传智播客旗下高端IT教育品牌