

# 1. 移动端基础

## 1.1. 浏览器现状

国内的UC和QQ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

## 1.2. 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：480x480,480x845,540x960,720x1080,1080x1920等

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**

## 1.3. 常见移动端屏幕尺寸

## 1.4. 移动端调试方法

- ChromeDevTools（谷歌浏览器）的模拟手机调试
- 搭建本地的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

# 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

一般移动端设备的浏览器都默认设置了一个布局视口，用于解决早期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上的网页大多都能再手机上呈现,只不过元素看上去很小,一般默认可以通过手动缩放网页

## 2.2. 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

## 2.3. 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,我们的布局的视口就有多宽

## 2.5. meta视口标签

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

## 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是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

# 3. 二倍图

## 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iPhone6/7/8是750\*1334

我们开发的时候1px不是一定等于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，称为物理像素比或者屏幕像素比

PC端和早期的手机屏幕、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，可以把更多的物理像素点压缩至一块屏幕内，从而达到更高的分辨率，并提高显示的细腻程度

## 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iPhone8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的需求

**背景图片注意缩放问题**

## 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

# 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）
- (2) 响应式页面兼容移动端（其次）

## 4.2. 单独制作移动端页面(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，如果是移动端打开，则跳到移动端页面

## 4.3. 响应式兼容PC移动端

三星电子官网，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的问题

# 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css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

## 5.2. 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
- 修复了浏览器的bug
- 是模块化的
- 拥有详细的文档

## 5.3. css3盒子模型box-sizing

传统模式宽度计算：盒子的宽度=css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

语法:

```
/*css3盒子模型*/  
box-sizing:border-box;  
/*传统盒子模型*/  
box-sizing:content-box;
```

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

## 5.4. 特殊样式

```
/* css3盒子模型 */  
border-sizing:border-box;  
-webkit-box-sizing:border-box;  
/* 点击高亮我们需要清除 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上系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;}
```

# 6. 移动端常见布局

移动端技术选型

移动端布局和以前学习的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

## 6.1. 流式布局（百分比布局）

流式布局，也称百分比布局，非固定像素布局

通过设置盒子的宽度为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)

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
- 注意代码里面的background-size也要写：精灵图原来宽度的一半