## A、移动Web开发准备

### 1、适配解决方案

流式布局 + viewport

### 2、流式布局

流式布局就是百分比布局，通过盒子的宽度设置成百分比(100%)来根据屏幕的宽度来进行伸缩，特点：不受固定像素的限制，内容向两侧填充。

流式布局是移动web开发使用的常用布局方式

### 3、Viewport

在移动端用来承载网页的这个区域，就是我们的视觉窗口，也叫viewport（视口）, 这个区域可是设置高度宽度，可是按比例放大缩小，而且能设置是否允许用户自行缩放。

### 4、Viewport的参数

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: 是否用户缩放，yes或no（1或0）



### 5、(meta:vp+tab)构建一个标准的移动Web开发页面

用meta标签把viewport的宽度设为device-width，同时initial-scale=1，user-scalable = 0就构建了一个标准的移动web页面

## B、移动端初始化

### 1、点击高亮效果

-webkit-tap-highlight-color : transparent;/\*清除点击高亮效果\*/

在移动端浏览器会遇见点击出现高亮的效果，在某项项目是不需要这个默认的效果的。那么我们通常会把这个点击的颜色设置成透明。

### 2、所有盒子以边框开始计算

/\*设置宽度以边框开始计算\*/  
-webkit-box-sizing: border-box;  
box-sizing: border-box;

在移动端通常使用的是百分比布局，那么这样的布局如果使用border或者padding或使容器的宽度超出屏幕的宽度产生滚动条。那么我们的解决方案是什么试用css3属性 box-sizing设置所有的盒子重边框开始计算宽度。

### 3、Input清除默认的样式

/\*在移动端怎么清除默认的样式\*/  
/\*在移动端清除浏览器默认样式\*/  
-webkit-appearance: none;

在移动设备的浏览器当中表单一般会有默认的属性 通过border：none

outline：none是无法完全清楚的，还是会有一些浏览器默认的属性，比如：

内阴影，立体感、、、向这些浏览器默认加上的样式用-webkit-appearance 这 个属性指的是设置成 none去除

## C、移动端常见问题

### 1、头部固定方式：

父盒子相对于body绝对定位（fixed）及width：100%，

子盒子的大盒子设置最大宽度，最小宽度以及margin：0 auto

### 1、最小宽度和最大宽度的限制

max-width: 640px; /\* 在行业当中的移动端的设计图一般使用的是640px\*/  
min-width: 320px; /\*在移动设备当中现在最小的尺寸320px\*/

适用：图片比较多的首页，门户，电商 等。

作用 保证页面在尺寸比较大的设备当中保证页面的效果也就是清新度

保证页面在小尺寸的设备当中有较好的布局效果。

### 2、Img的下间隙问题

<div>

abcdefghijklmnopqrstuvwxyz

<img src="../images/nv-fy.jpg" alt=""/>

</div>

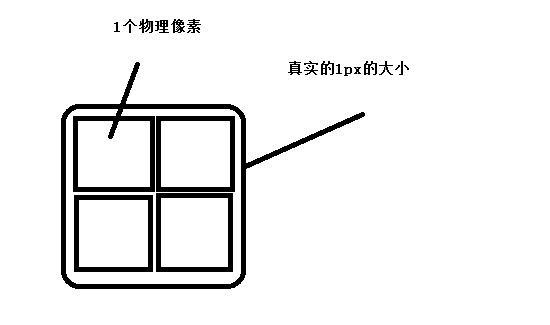
文字基线默认的 baseline 是以X的下边开始的

Img是行内块级元素 它也会有默认的基线对齐。那么和文字一样也会距离底部有一定的间隙。

### 3、图片模糊的原因以及解决方法

1：1的显示在移动设备当中图标会失真原因是什么如何解决？

在高清屏当中会用两个或多个物理像素来显示实际的1px图片内容



那么其实就是相当于把1px的图片放大显示了，所以有毛边的图片一般都会失真，也就是显示模糊。

解决方案，采用压缩图标尺寸的方式来解决。

如果是Img使用直接设置宽高的方式来压缩。

如果是背景使用的是设置background-size的方式来压缩

### 4、搜索按钮调用

<!--在移动端点击弹出输入法 enter键会显示搜索-->  
<form action="#">  
 <input type="search" placeholder="提示"/>  
</form>

search定义用于输入搜索字符串的文本字段。

在移动端调用输入法的时候会弹出小键盘，键盘一般是enter键，那么在搜索框当中我们要求调用是搜索按钮，那么这样的结构才能调用出来。

### 5、结构性伪类选择器

.sk\_product > li > p:nth-of-type(1){  
 color: #d8505c; }  
.sk\_product > li > p:nth-of-type(2){  
 /\*删除线\*/  
 text-decoration:line-through;  
 color: #666;  
}

E:first-of-type匹配同类型中的第一个元素E。

E:last-of-type匹配同类型中的最后一个元素E。

E:nth-of-type(n) 匹配同类型中的第n个元素E。

### 6、全屏单页面布局

使用百分比的方式：

*/\*满屏\*/***html**,**body**{  
 height: 100%;  
}

### 7、两栏其中一栏宽度自适应

**1、文本环绕的的概念：**

固定栏：浮动（float：left） 自适应栏：overflow：hidden

**（浮动的代码放在自适应代码的前面）**



那么当文本在另一个容器当中，容器是overflow：hidden的时候。那么这个时候这个盒子就变成了一个绝缘的盒子，不去影响任何外部的元素并且内容使用剩余的宽度。

1. **使用通栏式布局的方式实现宽度自适应（width:100% padding: n px）**

### 8、按钮问题



这里的按钮设置的是 40\*44的大小 但是内容只有20px为什么要这么设置呢？

在移动端如果按钮太小 触发起来相对来说比较困难，用户体验非常不好。那么在设置按钮的时候需要设置的更大一点。注意不能使用margin，应为a标签触发不了。

### 9、轮播图超出位置设置：

宽度：大盒子100% ul1000% li10% img100% 移动第一张到图片：transform:translateX(-10%)

### 10、结构类似的样式：

先写好一个统一的样式，然后定义不同的类名，在需要变化的地方添加相应的类名即可

## D、移动端事件

### 1.1、Touch事件(只存在于移动端)

**touchstart：当手指触碰屏幕时候触发。**

绑定事件的方法：

dom.addEventListener(**'touchstart'**,**function**(e){});

事件返回的e对象包含那些移动端特有的属性：

targetTouches 目标元素的所有当前触摸

changedTouches 页面上最新更改的所有触摸

touches 页面上的所有触摸

**touchmove：当手指在屏幕上滑动时连续触发。**

绑定事件的方法：

dom.addEventListener(**'touchmove'**,**function**(e){});

事件返回的e对象包含那些移动端特有的属性：

originalEvent  是jquery 封装的事件。

targetTouches 目标元素的所有当前触摸

changedTouches 页面上最新更改的所有触摸

touches 页面上的所有触摸

**touchend：当手指离开屏幕时触发。**

绑定事件的方法：

dom.addEventListener(**'touchend'**,**function**(e){});

事件返回的e对象包含那些移动端特有的属性：

changedTouches 页面上最新更改的所有触摸

**touchcancel：系统停止跟踪触摸时候会触发。**

这个事件不会经常使用，了解即可。

注意：在touchend事件的时候event只会记录changedtouches

clientX:触摸目标在视口中的X坐标。

clientY:触摸目标在视口中的Y坐标。

pageX：触摸目标在页面中的x坐标。

pageY：触摸目标在页面中的y坐标。

screenX:触摸目标在屏幕中的x坐标。

screenY:触摸目标在屏幕中的y坐标。

### **1.2、移动端的滑动效果**

在移动端特有的事件touch中，

包含了 touchstart，touchmove，touchend三个事件，

首先：我们可以通过touchstart事件中返回的event对象中的第一个触摸点信息中的client坐标，

然后：我们再通过监听touchmove事件的时候获取到滑动的时候的触摸点，在互获取event对象中返回的client坐标，这样就可以计算滑动的时候改变的距离。 同时改变当前元素的translate就可以。

最后：在touchend的时候利用记录下当前元素的定位。

注意的是：在移动端是利用transform来做定位。

滑动的时候改变距离的方向问题。

### 2、过渡和动画结束事件

**transitionEnd 过渡结束后触发。**

绑定事件的方法：

dom.addEventListener(**'webkitTransitionEnd'**,**function**(e){ });

dom.addEventListener(**'transitionEnd'**,**function**(e){ });

**animationEnd 动画结束后触发。**

绑定事件的方法：

dom.addEventListener(**'webkitAnimationEnd'**,**function**(e){ });

dom.addEventListener(**'animationEnd'**,**function**(e){ });

在这里要注意的是：为了兼容浏览器在绑定事件的时候需要同时绑定一个带webkit前缀的事件和不带的这样达到兼容主流浏览器。

还有：给目标元素绑定事件后，每一次过渡结束都会触发transitionEnd事件。

### 3、Gesture 事件（不常用）

gesturestart

当一个手指触摸屏幕之后，第二个手指再触摸屏幕时触发。

gesturechange

当上面的事件触发后立即触发。

gestureend

第二根手指离开屏幕时触发，之后将不会再次触发gesturechange。

在event当中会返回另外两个参数

scale 根据两个手指的滑动距离计算的缩放比例 初始1

rotation根据两个手指的滑动距离计算的旋转角度 初始 0

### 4、移动端事件原理

在移动端通常 tap swipe 等事件的封装原理

Tap：由于在移动端click会有300ms左右的延时，为了响应速度更快移动端框架一般会封装一个加tap的事件，原理：利用touch事件当没有滑动过并且响应时间在一定的时间内（比click快）那么这样的一个过程就是一个tap事件。

Swipe：在移动端有手势事件，也是一些移动端框架封装出来的那么这种事件也是touch事件开始的位置和结束的位置来判断手势的情况。是向下滑还是向右滑，是向下滑还是向上滑。

### 推荐网站

**动画网站**：<https://daneden.github.io/animate.css/>

这个网站可以提供一些常用的动画，可以在做移动开发的时候可以参考这个网站的效果完成动画。

**Zepto**：<http://zeptojs.com/>

Zepto是一个轻量级的针对现代高级浏览器的JavaScript库， 它与jquery有着类似的api。 如果你会用jquery，那么你也会用zepto。

**Weiui**：<http://weui.github.io/weui/>

微信UI库，它是微信前端团队提供给广大的微信开发者的一些ui组件库。方便开发这开发出和微信风格一致的 内嵌应用。

### 项目三种布局分析

首页的适用场景

门户电商的首页 而且内容比较丰富

分类页面的使用场景

是单页面 全屏的布局当中会使用

异步的交互 ajax web app 的开发模式当中会使用到

Web app 网页程序 网页应用

移动web页面优点： 跨平台 android ios

Hybird app 现在流行起来的 混合应用 开发模式

套用 嵌入 移动web页面的 应用所有的移动页面都可以

Native app 原生应用

购物车页面 任何地方都可以用到这样的布局。