1.媒介查询

主要是通过查询设备的宽度来执行不同的 css 代码，最终达到界面的配置。核心语法是： @media screen and (max-width: 600px) {

/\*当屏幕尺寸小于600px时，应用下面的CSS样式\*/ /\*你的css代码\*/

}

@media screen and (min-width: 375px){

html {

font-size: 14.0625px;

}

}

@media screen and (min-width: 360px){

html {

font-size: 13.5px;

}

}

@media screen and (min-width: 320px){

html {

font-size: 12px;

}

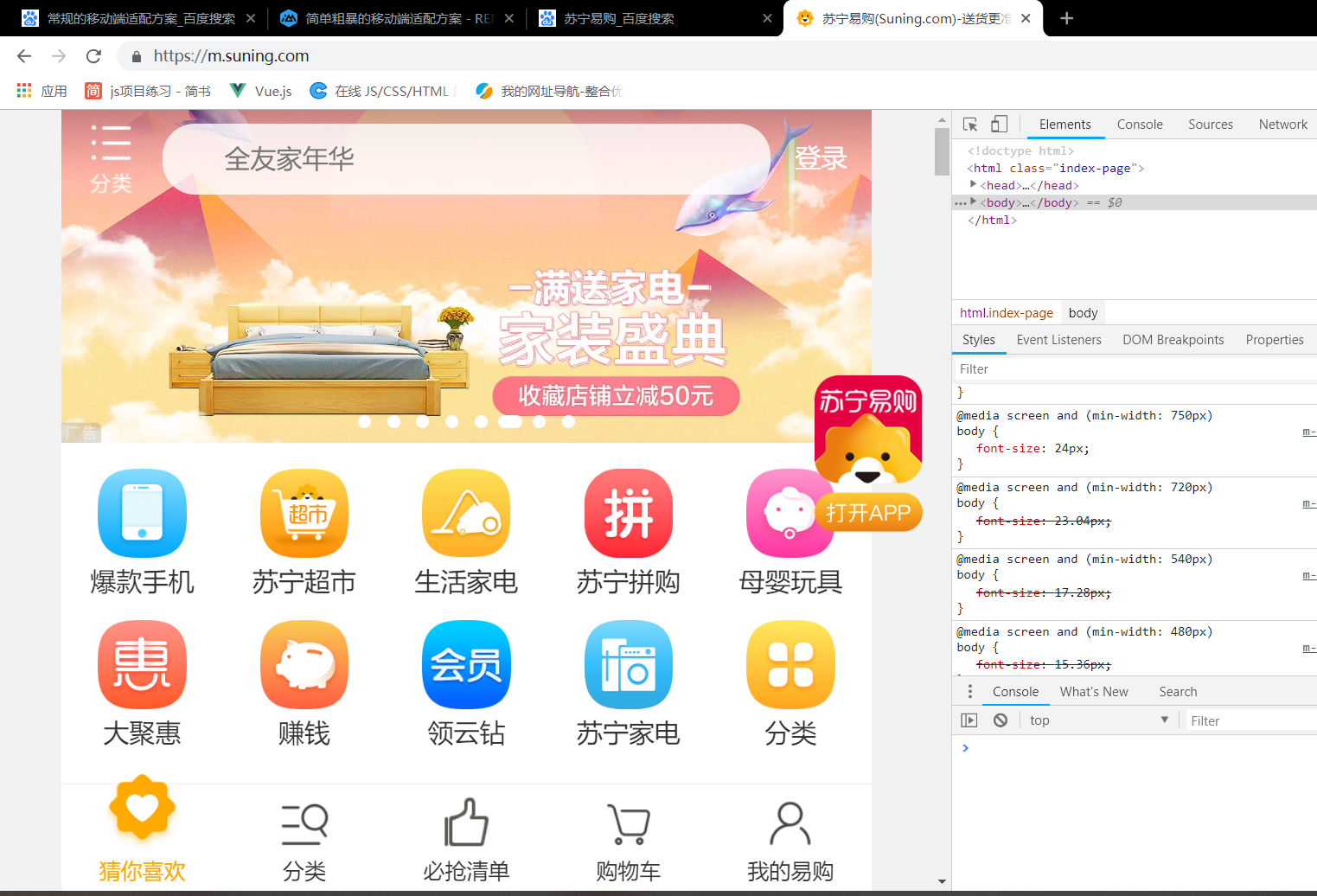
}

html {

font-size: 16px;

}

例如苏宁易购等网站均是这样做的



优点

media query可以做到设备像素比的判断，方法简单，成本低，特别是对移动和PC维护同一套代码的时候。目前像Bootstrap等框架使用这种方式布局

图片便于修改，只需修改css文件

调整屏幕宽度的时候不用刷新页面即可响应式展示

缺点

代码量比较大，维护不方便

为了兼顾大屏幕或高清设备，会造成其他设备资源浪费，特别是加载图片资源

为了兼顾移动端和PC端各自响应式的展示效果，难免会损失各自特有的交互方式

2.JS计算font-size

拿网易来说，它的设计稿应该是基于iphone4或者iphone5来的，所以它的设计稿竖直放时的横向分辨率为640px，为了计算方便，取一个100px的font-size为参照，那么body元素的宽度就可以设置为width: 6.4rem，于是html的font-size=deviceWidth / 6.4。这个deviceWidth就是viewport设置中的那个deviceWidth。根据这个计算规则，可得出不同分辨率下html的font-size大小如下：

deviceWidth = 320，font-size = 320 / 6.4 = 50px

deviceWidth = 375，font-size = 375 / 6.4 = 58.59375px

deviceWidth = 414，font-size = 414 / 6.4 = 64.6875px

deviceWidth = 500，font-size = 500 / 6.4 = 78.125px

这个deviceWidth通过document.documentElement.clientWidth就能取到了，所以当页面的dom ready后，做的第一件事情就是：

document.documentElement.style.fontSize = document.documentElement.clientWidth / 6.4 + 'px';

下面总结下网易的这种做法：

（1）先拿设计稿竖着的横向分辨率除以100得到body元素的宽度：

如果设计稿基于iphone6，横向分辨率为750，body的width为750 / 100 = 7.5rem

如果设计稿基于iphone4/5，横向分辨率为640，body的width为640 / 100 = 6.4rem

（2）布局时，设计图标注的尺寸除以100得到css中的尺寸，比如下图：



播放器高度为210px，写样式的时候css应该这么写：height: 2.1rem。之所以取一个100作为参照，就是为了这里计算rem的方便！

（3）在dom ready以后，通过以下代码设置html的font-size:

document.documentElement.style.fontSize = document.documentElement.clientWidth / 6.4 + 'px';

6.4只是举个例子，如果是750的设计稿，应该除以7.5。

（4）font-size可能需要额外的媒介查询，并且font-size不能使用rem，如网易的设置：

@media screen and (max-width:321px){

.m-navlist{font-size:15px}

}

@media screen and (min-width:321px) and (max-width:400px){

.m-navlist{font-size:16px}

}

@media screen and (min-width:400px){

.m-navlist{font-size:18px}

}

最后还有2个情况要说明：

第一，如果采用网易这种做法，视口要如下设置：

<meta name="viewport" content="initial-scale=1,maximum-scale=1, minimum-scale=1">

第二，当deviceWidth大于设计稿的横向分辨率时，html的font-size始终等于横向分辨率/body元素宽：

640\*680

641\*680

之所以这么干，是因为当deviceWidth大于640时，则物理分辨率大于1280（这就看设备的devicePixelRatio这个值了），应该去访问pc网站了。事实就是这样，你从手机访问网易，看到的是触屏版的页面，如果从pad访问，看到的就是电脑版的页面。如果你也想这么干，只要把总结中第三步的代码稍微改一下就行了：

var deviceWidth = document.documentElement.clientWidth;

if(deviceWidth > 640) {deviceWidth = 640;}

document.documentElement.style.fontSize = deviceWidth / 6.4 + 'px';

3. 淘宝的做法

看看淘宝在不同分辨率下，呈现的效果：



淘宝的效果跟网易的效果其实是类似的，随着分辨率的变化，页面元素的尺寸和间距都相应变化，这是因为淘宝的尺寸也是使用了rem的原因。在介绍它的做法之前，先来了解一点关于viewport的知识，通常我们采用如下代码设置viewport:

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">

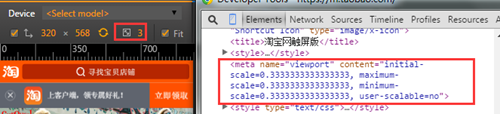
这样整个网页在设备内显示时的页面宽度就会等于设备逻辑像素大小，也就是device-width。这个device-width的计算公式为：

设备的物理分辨率/(devicePixelRatio \* scale)，在scale为1的情况下，device-width = 设备的物理分辨率/devicePixelRatio 。

devicePixelRatio称为设备像素比，每款设备的devicePixelRatio都是已知，并且不变的，目前高清屏，普遍都是2，不过还有更高的，比如2.5, 3 等。淘宝触屏版布局的前提就是viewport的scale根据devicePixelRatio动态设置：



在devicePixelRatio为2的时候，scale为0.5



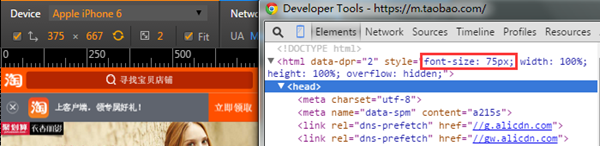
在devicePixelRatio为3的时候，scale为0.3333

这么做目的当然是为了保证页面的大小与设计稿保持一致了，比如设计稿如果是750的横向分辨率，那么实际页面的device-width，以iphone6来说，也等于750，这样的话设计稿上标注的尺寸只要除以某一个值就能够转换为rem了。通过js设置viewport的方法如下：

var scale = 1 / devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

淘宝布局的第二个要点，就是html元素的font-size的计算公式，font-size = deviceWidth / 10：



接下来要解决的问题是，元素的尺寸该如何计算，比如说设计稿上某一个元素的宽为150px，换算成rem应该怎么算呢？这个值等于设计稿标注尺寸/该设计稿对应的html的font-size。拿淘宝来说的，他们用的设计稿是750的，所以html的font-size就是75，如果某个元素时150px的宽，换算成rem就是150 / 75 = 2rem。总结下淘宝的这些做法：

（1）动态设置viewport的scale

var scale = 1 / devicePixelRatio;

document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');

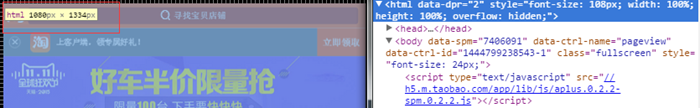
（2）动态计算html的font-size

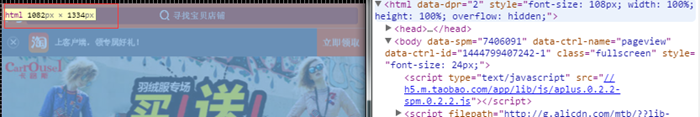
document.documentElement.style.fontSize = document.documentElement.clientWidth / 10 + 'px';

（3）布局的时候，各元素的css尺寸=设计稿标注尺寸/设计稿横向分辨率/10

（4）font-size可能需要额外的媒介查询，并且font-size不使用rem，这一点跟网易是一样的。

最后还有一个情况要说明，跟网易一样，淘宝也设置了一个临界点，当设备竖着时横向物理分辨率大于1080时，html的font-size就不会变化了，原因也是一样的，分辨率已经可以去访问电脑版页面了。





阿里团队高清方案布局代码，放到head标签即可，注意不要设置meta viewport，因为会自动生成

<script>!function(e){function t(a){if(i[a])return i[a].exports;var n=i[a]={exports:{},id:a,loaded:!1};return e[a].call(n.exports,n,n.exports,t),n.loaded=!0,n.exports}var i={};return t.m=e,t.c=i,t.p="",t(0)}([function(e,t){"use strict";Object.defineProperty(t,"\_\_esModule",{value:!0});var i=window;t["default"]=i.flex=function(e,t){var a=e||100,n=t||1,r=i.document,o=navigator.userAgent,d=o.match(/Android[\S\s]+AppleWebkit\/(\d{3})/i),l=o.match(/U3\/((\d+|\.){5,})/i),c=l&&parseInt(l[1].split(".").join(""),10)>=80,p=navigator.appVersion.match(/(iphone|ipad|ipod)/gi),s=i.devicePixelRatio||1;p||d&&d[1]>534||c||(s=1);var u=1/s,m=r.querySelector('meta[name="viewport"]');m||(m=r.createElement("meta"),m.setAttribute("name","viewport"),r.head.appendChild(m)),m.setAttribute("content","width=device-width,user-scalable=no,initial-scale="+u+",maximum-scale="+u+",minimum-scale="+u),r.documentElement.style.fontSize=a/2\*s\*n+"px"},e.exports=t["default"]}]); flex(100, 1);</script>

代码原理

这是阿里团队的高清方案布局代码，所谓高清方案就是根据设备屏幕的DPR（设备像素比，又称DPPX，比如dpr=2时，表示1个CSS像素由4个物理像素点组成）

动态设置 html 的font-size, 同时根据设备DPR调整页面的缩放值，进而达到高清效果。

有何优势

引用简单，布局简便

根据设备屏幕的DPR,自动设置最合适的高清缩放。

保证了不同设备下视觉体验的一致性。（老方案是，屏幕越大元素越大；此方案是，屏幕越大，看的越多）

有效解决移动端真实1px问题（这里的1px 是设备屏幕上的物理像素）

比较网易与淘宝的做法

共同点：

都能适配所有的手机设备，对于pad，网易与淘宝都会跳转到pc页面，不再使用触屏版的页面

都需要动态设置html的font-size

布局时各元素的尺寸值都是根据设计稿标注的尺寸计算出来，由于html的font-size是动态调整的，所以能够做到不同分辨率下页面布局呈现等比变化

容器元素的font-size都不用rem，需要额外地对font-size做媒介查询

都能应用于尺寸不同的设计稿，只要按以上总结的方法去用就可以了

不同点

淘宝的设计稿是基于750的横向分辨率，网易的设计稿是基于640的横向分辨率，还要强调的是，虽然设计稿不同，但是最终的结果是一致的，设计稿的尺寸一个公司设计人员的工作标准，每个公司不一样而已

淘宝还需要动态设置viewport的scale，网易不用

最重要的区别就是：网易的做法，rem值很好计算，淘宝的做法肯定得用计算器才能用好了 。

3.vh vw方式

1，vw、vh、vmin、vmax 的含义

（1）vw、vh、vmin、vmax 是一种视窗单位，也是相对单位。它相对的不是父节点或者页面的根节点。而是由视窗（Viewport）大小来决定的，单位 1，代表类似于 1%。

视窗(Viewport)是你的浏览器实际显示内容的区域—，换句话说是你的不包括工具栏和按钮的网页浏览器。

（2）具体描述如下：

vw：视窗宽度的百分比（1vw 代表视窗的宽度为 1%）

vh：视窗高度的百分比

vmin：当前 vw 和 vh 中较小的一个值

vmax：当前 vw 和 vh 中较大的一个值

2，vw、vh 与 % 百分比的区别

（1）% 是相对于父元素的大小设定的比率，vw、vh 是视窗大小决定的。

（2）vw、vh 优势在于能够直接获取高度，而用 % 在没有设置 body 高度的情况下，是无法正确获得可视区域的高度的，所以这是挺不错的优势。

3，vmin、vmax 用处

做移动页面开发时，如果使用 vw、wh 设置字体大小（比如 5vw），在竖屏和横屏状态下显示的字体大小是不一样的。

由于 vmin 和 vmax 是当前较小的 vw 和 vh 和当前较大的 vw 和 vh。这里就可以用到 vmin 和 vmax。使得文字大小在横竖屏下保持一致。

4，浏览器兼容性

（1）桌面 PC

Chrome：自 26 版起就完美支持（2013年2月）

Firefox：自 19 版起就完美支持（2013年1月）

Safari：自 6.1 版起就完美支持（2013年10月）

Opera：自 15 版起就完美支持（2013年7月）

IE：自 IE10 起（包括 Edge）到现在还只是部分支持（不支持 vmax，同时 vm 代替 vmin）

（2）移动设备

Android：自 4.4 版起就完美支持（2013年12月）

iOS：自 iOS8 版起就完美支持（2014年9月）