

基于响应式设计的 Web App 界面布局研究

李颖

(安徽工程大学 艺术学院,安徽 芜湖 241000)

摘要:为了解决 Web App 运行于多种设备屏幕和浏览器上的显示问题,提出 Web App 跨平台应用技术方案,以解决设备屏幕的自适应问题和浏览器兼容性问题;并提出基于响应式设计的界面布局方案,用以解决当设备屏幕和用户行为变化时,Web App 界面元素的定位和显示问题。其关键技术为基于弹性盒的缩放模型以及界面元素的定位方法,通过设计实践和发布测试实验验证了这一方法的有效性。基于此方案设计开发的 Web App 能够实现不同设备环境和用户使用行为下的合理布局与正确显示,始终带给用户良好的视觉体验。

关键词:跨平台;界面布局;Web App;浏览器;响应式设计

DOI: 10.13757/j.cnki.cn34-1328/n.2017.03.017

中图分类号:G420-057

文献标识码:A

文章编号:1007-4260(2017)03-0062-06

Interface layout study on web app based on responsive design

LI Ying

(School of Art, Anhui Polytechnic University, Wuhu 241000, China)

Abstract: In order to resolve the display problem while running on different screen of devices and different web browsers of Web App, the cross-platform application technology scheme is used to resolve the self-adaptive problem with different device screens, resolution and the compatibility issue with different web browsers. The interface layout scheme based responsive design is proposed to resolve the position and display problem caused by the change of device and user behave. The key technology is the scale model based on flexible box and the position method of the interface elements. And the effectiveness of this method is verified by design practice and the test experiment. Web App designed and developed by this method can realize a reasonable layout and correct display on different web browsers of many different devices, and bring users good visual experience all the time.

Key words: cross-platform; interface layout; Web App; web browser; responsive design

随着手机等移动设备的普及和网络技术的发展,移动端在线应用成为主流,其中较为流行的有 Native App, Hybrid APP 和 Web App。Native Application^[1]由于其使用前的下载和安装以及维护成本问题受到众多诟病,Hybrid APP 和 Web APP 的运行更多地依赖 HTML5 技术^[2],能够实现数据的在线更新、检索和分享,得到广泛应用与研究。作为运行于移动浏览器端的在线应用,响应式设计原理的运用能够较好地解决 Web App 在跨设备应用时界面正确显示问题。近年来,响应式 Web 设计被发展和应用,特别是 An-

droid 系统^[3]和苹果系统在响应式界面设计领域取得了很大的成功。本文在总结成功设计经验的基础之上,深入研究 Web App 在多种设备环境下的运行和显示问题,探索 Web App 跨平台应用技术以及跨设备应用时的自适应布局方案,以实现 Web App 在不同设备环境和用户使用行为下的合理布局与正确显示。

1 Web App 发展与特征描述

Web App 作为近年来较为流行的浏览器端在线应用,受到网络用户的认可以及国内外学者的

* 收稿日期:2016-12-28

基金项目:国家自然科学基金(51605003),和安徽工程大学青年科研基金(2016YQ06)和安徽工程大学引进人才科研启动基金(2015YQQ003)。

作者简介:李颖,女,安徽滁州人,硕士,安徽工程大学艺术学院讲师,研究方向为 Web 前端产品设计与开发。E-mail: ly408493042@163.com

关注。Serrano Nicolas 等^[4]提出了 Web App 的起源与概念,他们认为,最早的 Web App 是在电脑桌面的网页浏览器上运行的智能应用程序,随着手机等移动设备的普及和无线网络的发展,Web App 的概念推广至整个网络环境中的多种终端设备浏览器上运行的智能程序。Elgan Mike^[5]通过调查得出,手机用户 86%的时间花费在手机 App 上,而只有 14%的时间花费在我们所熟知的万维网上。他还提出大胆假设,这或许将掀起一场互联网的风暴革命,Web App 未来可能会取代 www 服务。致力于手机 Mobile App 开发的 ASANA^[6]公司研发出支持苹果 IOS 系统的团队协作 App^[7],可以脱离万维网和 email 独立工作。技术人员还研究了一种基于顾客体验的餐馆信息服务的 DIY 在线应用^[8]。这预示着在未来社会,Web App 可能会逐步取代万维网,成为未来网络用户获取网络服务的最重要手段。

作为一种无需下载的在线应用,Web App 拥

有与 NativeAPP 相媲美甚至超越 Native APP 的用户体验,同时具备可被检索与智能分发的特性,因而受到 Web 用户的追捧。由于其基于“云端”服务的技术平台,Web APP 具有模块化应用生成、大数据分析功能、跨平台应用和丰富的检索和推广体验等技术特征。用户体验方面,Web APP 具有轻便、快捷、有效和丰富的交互体验等特点,能够提供良好的用户体验^[9]。

2 Web App 的跨平台应用技术

Web App 实现跨平台跨设备应用的前提是解决设备屏幕的自适应问题和网页浏览器(如谷歌 chrome、火狐 firefox、opera、Safari、IE 等)的兼容性问题。核心技术方案为基于 HTML5 的跨平台应用技术,具体包括以下几个部分。

2.1 设备屏幕与分辨率的自适应问题

多种移动设备具有不同的设备属性与屏幕分辨率,表 1 列举了几款主流手机的相关参数。

表 1 几款主流手机相关参数

机型	三星 s6	魅蓝 Note5	华为荣耀 v8	小米 5s	红米 Note4	iphone6	iphone6 plus	iphone6s
系统平台	Android5.0	Flyme5	Android6.0	MIUI8	MIUI8	iOS8.0	iOS8.0	iOS9.0
屏幕尺寸	5.1 英寸	5.5 英寸	5.7 英寸	5.15 英寸	5.5 英寸	4.7 英寸	5.5 英寸	4.7 英寸
分辨率	1 440*2 560 px	1 920*1 080 px	1 920*1 080 px	1 920*1 080 px	1 920*1 080 px	1 334*750	1 920*1 080 px	2 000*1 125 px
dpi	576	401	515	428	401	326	400	488

鉴于多款手机不同的设备属性与屏幕分辨率,Web App 界面要正确显示于多种设备屏幕首先要解决设备屏幕的自适应问题。基于 HTML5 的编辑方式,可在网页头文件中添加相关属性以控制其显示方式。以 iphone 系统的 safari 浏览器为例,其浏览器显示区域为 980px,显示方式可以设置为:

(1)以原始大小显示,不允许缩放

若 Web App 界面布局固定,缩放会影响界面预设的视觉效果,可采用以原始大小显示界面内容,在<head>中添加如下属性:

```
<meta name='viewport' initial-scale=1.0
minimum-scale=1.0 maximum-scale=1.0
user-scalable=no>
```

其中 initial-scale,minimum-scale,maximum-scale 控制其缩放比例为原始大小,user-scalable 属性设置缩放模式为“无”。

(2)可视区调整屏幕显示

默认情况下,iphone 的 safari 会像在大屏幕的桌面浏览器那样显示页面,达 980px,然后自动缩小内容以适应 iphone 的小屏幕。此时应用界面内容比较繁杂,用户视觉体验效果非常不好:字体太小、模糊、图片不清晰等。为了达到较为人性化的视觉效果,可对页面可视区域进行设置:

```
<meta name='viewport'
content='width=device-width'>
```

Content 属性内容通知浏览器使用设备宽度作为可视区域,从而适度放大界面显示区域使界面内容的呈现更加直观和清晰。

(3)手动设置可视区域

若 Web App 界面宽度较小,如 750px,在 iphone 中浏览时会看到 980px 剩余的空间充满白色,严重影响界面的整体视觉效果,为了解决这一问题,可以手动设置页面可视区域的显示宽度:

```
<meta name='viewport' content='width=
980px'>
```

以上 3 种显示设置方案针对 Web App 用户界

面设计的不同情况,而对于大多数 Web 应用程序,普适性较高的一种设置方案为:

```
<meta name='viewport'
content='width=device-width'
initial-scale=1.0 user-scalable=no >
```

显示区域设定为屏幕最大宽度,禁止用户缩放。

2.2 设备浏览器兼容性问题

随着移动设备和跨平台网络应用技术的推广,HTML5 提供了一个普遍的 Web 标准,使得网络应用程序能够广泛运行于不同的设备浏览器之上。相比于传统桌面 Web 开发,基于 HTML5 可以方便的构建类似传统客户端软件的浏览器 App,可以访问磁盘系统和摄像头等敏感设备,轻松将桌面软件擅长的领域带入 Web 世界^[10]。移动 Web App 能够实现兼容电脑、手机、iPad 等多种设备平台和浏览器的应用。

2.2.1 Web 新增属性的兼容性问题

目前网络市场的主流浏览器有微软的 IE、苹果的 Safari、Firefox、Opera、谷歌的 Chrome 等。由于各浏览器开发商的专利保护问题,不同公司推出的浏览器具有不同的内核。而不同浏览器对 HTML5 及 CSS 3.0 新增的元素属性都有不同的支持,通过添加前缀以兼容主流浏览器,如 chrome 浏览器-webkit-,火狐浏览器-moz-,IE 浏览器-ms-,opera 浏览器-o-。兼容多种浏览器就要添加多个前缀。以渐变为例,通过添加-webkit-前

缀以兼容谷歌浏览器:

```
background:-webkit-linear-gradient(top,black,
white)
```

2.2.2 多媒体资源播放的兼容性问题

由于编码和解码制式的问题,不同浏览器对视音频文件格式的支持情况不同,表2展示了主流浏览器对视音频格式的支持情况。可以看出,主流浏览器对音频格式的支持情况都较好,而对于编码制式更为复杂的视频格式,支持情况有所不同。

为了更好地在页面中引用视音频格式,HTML5 使用<video>或<audio>标签引用视音频文件。使用<source>标签来引用多种格式的视音频文件以适应苹果、安卓等主流设备视音频播放控件。如下:

```
<video controls>
<source src='myVi1.webm' type='video/
webm'>
<source src='myVi1. ogg' type='video/
ogg'>
<source src='myV1. mp4' type='video/
mp4'>
.....
</video>
```

不同浏览器在 video 和 audio 元素上对视音频编码格式尚无统一的标准,但通常而言,表2中所列举的几种视音频格式基本上可以覆盖现代主流浏览器。

表2 主流浏览器对视音频格式的支持情况

	WebM	Ogg	MP4	MPEG H.264	MP3 或 WAVE
chrome	6.0 以上版本支持	5.0 以上版本支持	5.0 以上版本支持	原生支持	支持
Firefox	4.0 以上版本支持	3.5 以上版本支持	不支持	需第三方解码器	支持
Safari	需安装插件	不支持	3.0 以上版本支持	原生支持	支持
IE	需安装插件	不支持	9.0 以上版本支持	原生支持	支持
Opera	10.6 以上版本支持	10.5 以上版本支持	不支持	不支持	支持

2.2.3 Web App 界面的响应式设计

以上方案解决了设备屏幕尺寸及分辨率的自适应问题以及浏览器的兼容性问题,接下来的问题是当设备屏幕和浏览器宽度改变时,界面元素如何布局才能保证其在网页上的正确显示。其解决方案为响应式界面设计,以一款 jQuery 图像切换 Web App 为例逐一剖析在作响应式界面设计时可能遇到的具体问题及其解决方案。

3 基于响应式设计的界面布局

对于给定界面宽度的 Web App,当遇到不同大小的屏幕分辨率和浏览器,会整体缩放以适应屏幕的变化,但这一改变有时会带来用户视觉上的不良感受,如元素因缩放拉伸而改变了原先的宽高比,或缩小后的字体模糊、图片观感不好等。

响应式设计正是为了解决设备环境改变后界面及其元素的正常显示问题。Marcotte 早在 2010 年就提出了响应式网页设计的概念^[11]。响应式网页设计要求网页布局和内容根据设备与

用户使用行为的变化而变化,用于构建交互性强的动态Web界面。

响应式Web设计的关键是布局问题,其解决方案为基于CSS3弹性盒模型结合多种定位方式(浮动定位、相对定位、绝对定位等)的自适应界面布局方案。

其核心方法是网页及元素的宽度均使用百分比以适应不同设备屏幕的宽度;部分元素采用弹性盒模型以适应屏幕的变化;需要固定大小的元素采用合理的定位方式保证其在界面中的位置关系。其中较为关键的技术为弹性盒技术和元素的定位方法。弹性盒布局模型专为不同尺寸和不同设备的元素排布而设计,可以说它就是为移动而生的强大技术。弹性盒模型可应用于界面、菜单、按钮等网页元素的布局。

3.1 弹性盒模型

弹性盒模型由伸缩容器(父元素)和伸缩项目(子元素)组成,通过设置元素的display属性为flex或inline-flex可以得到一个伸缩容器,伸缩容器包含若干伸缩子元素,弹性盒模型如图1所示。Flex容器有能力让其子元素改变宽度、高度(甚至顺序),以最佳方式填充可用空间,以适应所有类型的显示设备和屏幕大小,这是其核心思想。可以通过设置flex-flow的值控制子元素按主轴线伸缩(row)或者侧轴线伸缩(column),或者反向伸缩(row-reverse和column-reverse)。

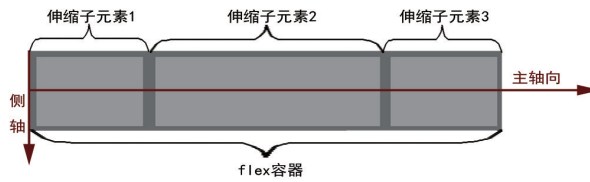


图1 弹性盒模型

3.2 基于弹性盒模型的响应式界面设计

响应式界面设计提供了针对不同设备和浏览器的灵活布局方案。通过弹性盒模型和CSS3定位方法实现网页、菜单、按钮的响应式布局;建议设计过程中,网页宽度与元素宽度以及位置尽量使用百分比而非具体数值,以适应不同设备屏幕大小。

3.2.1 界面整体布局

(1)优先显示。作为基于网页的智能应用程

序,Web App在网页中的位置直接影响到用户的视觉效果和使用效果。一般而言,为了使Web App凸显于页面重要位置(如淘宝首页幻灯),且在页面加载时优先出现于用户视线(即优先渲染),可采用“双飞翼布局”,如图2所示。左右内容宽度固定,App内容铺满屏幕的剩余宽度,从而自动适应屏幕大小的变化。且App部分在网速较慢的情况下能够优先渲染显示其内容。

(2)平行显示。当App内容作为网页辅助或者平行应用时(如淘购物车),可将其与其他网页内容平行显示,此时可采用“栅格布局”,如图3所示。为页面的不同应用设置宽度百分比(数值依实际情况而定),当显示屏幕变化时,网页各部分内容自动调整,且宽度分布按百分比变化。

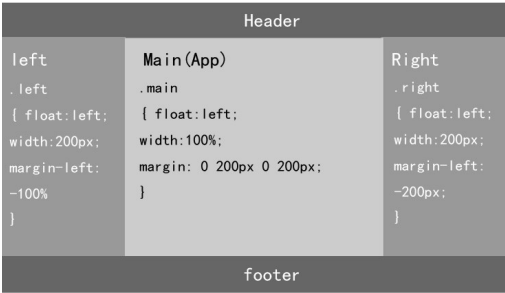


图2 双飞翼布局

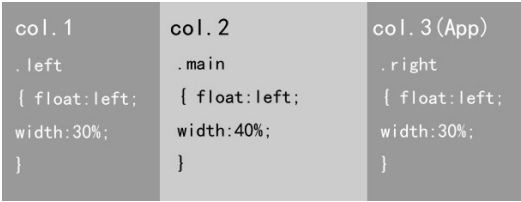


图3 栅格布局

3.2.2 界面元素定位

(1)容器与元素定位。App界面元素(文本、图片、视频、音频、超链接等)的响应式布局涉及定位问题,具体而言是采用何种定位方法:相对定位(relative)、绝对定位(absolute)还是浮动定位(float)。一般而言,Web App的框架容器采用相对定位(relative)以适应设备屏幕的变化;App元素根据需要进行选择其定位方式。

(2)边距参数。根据实际情况,需要与容器边距保持固定距离的元素,边距给予固定数值,如#btnLeft{position: absolute; left:0px;top:185 px; }

需要保持相对位置关系(如居中、偏左和偏右)的元素边距给予百分比,如如下定位的logo元素大约始终保持在界面中央位置:

```
#main.logo{position:absolute;bottom:50%;
```

right: 50%;}

当然百分比边距也要根据元素的大小等进行适当调整,例如元素本身占有一定宽度而又需要保持居中位置时,宽度计算时需要减去元素宽度的一半:位置%=1/2(屏幕%-元素宽度%)。

(3)浮动和对齐。为了保持界面的整体美观,需要在视觉效果上保持对齐效果。align属性用以解决元素的对齐问题,包括left\centre\right等。而浮动对齐方式(float)作为一种元素活动定位方式,根据其左右元素的实际情况进行动态调整,是一种适合页面响应式布局的灵活排版方式。常用的浮动方式有向左浮动和向右浮动。

3.3 设计实例

基于以上布局方案设计的Web App界面能够根据屏幕宽度和用户操作的变化自动调整界面宽度和元素位置;部分元素,如基于弹性盒模型的菜单按钮等自动伸缩以适应界面宽度的变化;其他元素根据实际情况选择合适的定位方式。基于以上布局方案设计开发一款jQuery图片切换Web App:

(1)App界面整体采用双飞翼布局(未显示边框),页面加载时始终优先显示App部分;

(2)容器框架采用相对定位且宽度给定百分比随着屏幕宽度的变化而变化:

```
#main{width:85%; position:relative; }
```

```
#main .box{width:90%; position:relative; }
```

(3)图片采用相对定位方式且给定左上边距以保证其从容器的左上角位置开始显示:

```
#main .box img{width:100%; position: absolute; left: 0px;top:0px;}
```

(4)顶部菜单使用了弹性盒布局方法以实现根据界面宽度的变化自动伸缩:

```
#main #flex1{ width:100%; display: -webkit-flex;}
```

```
#main #flex1 a{ text-align:center; -webkit-flex:1; }
```

(5)左右箭头相对于容器外框绝对定位以保证其始终处于界面的边缘位置:

```
#btnLeft{ position: absolute; left:0px;top:185px}
```

```
#btnRight{position:absolute;right:0px;top:185px}
```

(6)底部翻页按钮相对定位且其位置给予百分

比以保证其始终处于界面容器的底部中央位置:

```
#main .page{width:132px;height:22px;
position:relative; top:-100px;left:43%;}
```

4 发布测试

最终将基于以上布局方案的jQuery图片切换Web App发布于谷歌浏览器的设备模拟器(谷歌浏览器带有较为全面的屏幕设备模拟器,包括iPad,iphone系列及其他移动端设备,也可以添加设备以备测试)。图4-7分别为jQuery图片切换Web App在PC(1366*736)浏览器以及谷歌浏览器设备模拟器的iphone 6 plus(横屏,736*414)、三星Galaxy Note3(横屏,640*360)、Apple iPad(1024*768)中的测试效果。

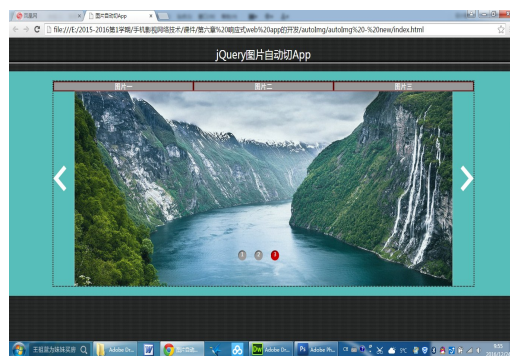


图4 PC端测试结果



图5 iphone 6 plus测试结果(横屏显示效果)

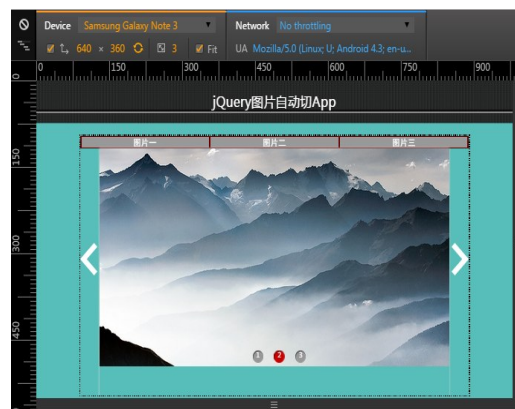


图6 三星Galaxy Note3测试结果(横屏显示效果)



图7 iphone iPad测试结果

从以上测试结果可以看出,基于响应式设计的Web App运行于不同设备终端时,能够响应屏幕宽度变化调整其容器框架及内容。其中黑色虚线为App的框架容器边框,这是一个虚拟的概念,无需在页面中显示出来,但为了便于理解,笔者在页面中用黑色虚线border显示出来。从测试结果可以看到,无论在何种设备中打开网页,都能优先显示App主体部分,且App界面始终占屏幕的给定宽度比(以上实例中是85%);采用弹性盒布局的顶部菜单能够根据界面宽度的变化而自动伸缩;进行绝对定位的左、右箭头始终处于界面的边缘位置;进行相对定位的翻页按钮始终处于界面底部中央位置,且保持各元素宽高比和大小不变,真正体现了响应式设计的核心思想。

5 结 论

作为基于浏览器端的在线应用,Web App运行于多种设备屏幕和浏览器端时,其界面显示因设备和用户操作而有所不同。设计实践和测试结果表明,基于HTML5的跨平台应用技术能够解决Web App界面适应多种设备屏幕和浏览器的显示问题。同时,基于响应式设计的界面布局

方案很好地解决了Web App在不同设备环境和用户使用行为下的界面正常显示问题。响应式设计的关键技术是基于弹性盒的缩放模型以及界面元素的灵活定位。基于此界面布局方案设计开发的Web App能够运行于不同设备环境,且无论用户使用行为如何变化,都能够正常合理地显示界面,始终带给用户良好的视觉体验效果。

参考文献:

- [1] 黄悦深. 基于HTML5的移动Web App开发[J]. 图书馆杂志, 2014, 133(7): 72-77.
- [2] 田嵩. 基于轻应用的移动学习内容呈现模式研究[J]. 电化教育研究, 2016(2): 31-37.
- [3] 莱特曼 (Juhan Lehtimäki). 精彩绝伦的Android UI设计: 响应式用户界面与设计模式[M]. 北京: 机械工业出版社, 2013: 30-41.
- [4] NICOLAS S, JOSUNE H, GORKA G. Mobile web apps[J]. IEEE Software, 2013, 30(5): 22-27.
- [5] MIKE E. Mobile apps bringing the end of the world wide web as we know it[EB/OL]. <http://www.eweek.com/mobile/mobile-apps-bringing-the-end-of-the-world-wide-web-as-we-know-it> [2014-04-10].
- [6] VINTON C. Apps and the web[J]. Communications of the ACM, 2016, 59(2): 7.
- [7] KONRAD A. Asana finally launches a native ios app after facebook cofounder bet wrong on web-based apps[EB/OL]. <https://www.forbes.com/sites/alexkonrad/2014/07/29/asana-finally-launches-a-native-ios-app/#7d7608b6664f> [2014-07-29].
- [8] MEENEGHAN K. Moving from mobile apps to web apps[J]. Restaurant Business, 2013, 112(4): 79.
- [9] HASSENZAHL M, TRACTINSKY N. User Experience a Research Agenda [J]. Behavior and Information Technology, 2006, 25(2): 91-97.
- [10] 林珑. HTML5移动Web开发实战详解[M]. 北京: 清华大学出版社, 2014: 14-17.
- [11] 冯春英. 基于响应式Web设计的新型图书馆门户网站构建[J]. 图书馆学研究, 2015, (15): 34-40.