

**本科毕业设计（论文）**

**基于SSM框架的顺丰优选网站的设计与实现**

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 | 姚煜 |
| 学 号 | 201317020229 |
| 指导教师 | 曹洪宇 |

二〇一七年五月十五日

摘 要

随着当今互联网的飞速发展，网上购物的需求量增加日益明显，虽然市面上也存在着诸如淘宝之类的网上购物平台，但物流的硬伤成为了众顾客心中的隐痛。所以我决定应用结合顺丰自有的物流资源，开发出顺丰优选网上购物平台。

本文主要阐述了使用前端最新技术和效果配合java完成顺丰优选官方网站的设计与实现过程。包括从可行性到网站发布的相关文档。本网站基于jQuery类库，前端使用Html5,CSS3,JavaScrip等，后台使用Jdbc，JavaServlet等相关技术，实现了主页，商品详情，商品列表，购物车，收货详情等页面。

由于本系统使用前端Html5一些新的元素和CSS3的新增属性完成页面的设计和实现，体现了创新。使用AJAX的技术来与后台进行数据交互，减少了页面响应时间。使得用户的体验效果更加美好。体现了电商网站最重要的一点，用户体验第一。

**关键字**：html5，AJAX，前端开发，用户体验

Abstract

**目 录**

摘要 **I**

**Abstract II**

第**1**章概述 **1**

1.1 引言 1

1.2 国内外现状研究 1

1.3 研究内容和拟解决问题 2

1.3.2 交互性 2

1.3.3 性能 2

1.4 本章小结 3

第**2**章方案论证 **4**

2.1 相关技术 4

2.1.1 HTML5 4

2.1.2 SSM 5

2.1.3 JavaScript/DOM/Ajax/JSON 5

2.1.4 网站重构 6

2.1.5 Web优化 6

2.1.6 Web 布局 6

2.2 可行性分析 7

2.2.1 技术可行性 7

2.2.2 操作可行性 7

2.2.3 经济可行性 8

2.3 结论 8

2.4 本章小结 8

第 **3** 章需求分析 **9**

3.1 需求分析概述 9

3.2 数据流分析 9

3.3 用例分析 10

3.4 功能分析 11

3.4.1 系统包图 12

3.4.2 系统类图 12

3.5 本章小结 12

第**4**章过程论述 **13**

4.1 概要设计 13

4.1.1 前端总体开发流程及设计 13

4.1.2 组织结构设计 14

4.1.3 前端总体布局和UI设计 14

4.1.4 数据库设计 16

4.1.5 程序流程图 19

4.1.6 时序图 20

4.1.7 活动图 21

4.2 详细设计 22

4.2.1 首页banner轮播图 22

4.2.2 首页楼层导航 23

4.2.3 商品展示页面 25

4.2.4 商品详情页面 25

4.2.5 购物车页面 29

4.2.6 购买商品 31

4.2.7 我的订单 33

4.2.8 管理收货地址 34

4.2.9 登陆注册 35

4.3 本章小结 36

第**5**章软件测试与结果分析 **37**

5.1 前端开发中常见Bug及修复 37

5.1.1 IE浏览器常见Bug 37

5.1.2 JavaScript兼容性Bug 42

5.1.3 CSS兼容性Bug 43

5.2 前端的测试方法及测试工具的使用 45

5.2.1 本地测试方式 45

5.2.2 测试开发工具 45

5.3 测试用例和测试结果 46

5.3.1 测试用例摘要 46

5.3.2 测试结果摘要 47

5.4 本章小结 49

结论 **50**

致谢 **51**

参考文献 **52**

附录 **54**

第1章 概述

1.1 引言

随着计算机的发展和普及,人民的生活变得越来越方便。互联网时代的来临为我们诸多方面带来了无限商机与便利。电子商务一词大家肯定不陌生,从我们的衣物、家居,再到生活用品。我们都可以通过各大电商网站提供的网络平台,点击购买自己喜欢的商品就可以了。极大地方便了我们的生活。对于电商网站来说,它只需要在网络上创建一个销售平台,将自己所有的商品展示在网络上。而用户可以查看、选购。而供货商只需要生产,然后通过物流便可以送到用户手中。没有中间商、零售商。这样一来商品就没有层层的中间商加价。这样用户就能以最低的价格获得自己心仪的商品，或者买到性价比最高的商品。节约成本,提高了速度,缩小了用户等待时间。

顺丰优选，它是一个能够买到全球美食的电商网站，以方便人们的生活，满足用户的需求为己任，为用户挑选优质的商品。

1.2 国内外现状研究

电子商务因其方便快速以及在用户和商家直接构建起一条方便快捷的渠道，得到广大用户的青睐从而迅速发展成为如今互联网最炙手可热的在线网站。

在购物服务领域，Amazon 是国外人们最喜爱的购物网站。它是美国最大的电子商务公司，也是美国最火的，也可以说是全球最大的电子商务网站。

目前它已在全球都有用网络平台来为用户提供购买服务。它从以前的全球第一书店成为现在涉及基本所有的商品行业。

而在中国，如今中国的物联网已经是世界一流了，我们不仅实现了“上午购物下午到家”，还普及了无现金的生活。提到购物网站，我们首先想起的一定是淘宝，淘宝是世界最大的购物网站，它现在的发展已经远超Amazon和eBay。

但顺丰优选也是一家在中国占有巨大销售量的电商网站。顺丰优选结合线上电商网站和线下的门店，以自己的物流为优势，不论用户需要的美食在何地，顺丰优选都可以提供。

本网站采用html5，CSS3,JavaScript，jQuery等前端技术和框架，结合subline text ，Chrome，eclipse,mysql等工具。设计和实现高效、美观、用户体现高的前端页面。目的在于提高和锻炼自己的前端设计能力，规范前端开发流程，提供工作效率。为以后从事前端开发行业打下一定的基础。同时探讨Web开发的基本内容、前端布局和交互设计、前端性能优化以及前端常见问题的处理方式、前端测试方法和前端测试工具等问题。

1.3 研究内容和拟解决问题

**1.3.1** 布局设计

一个好的网站，必须要有一个好的布局，使得用户使用起来既方便又美观，体现了良好的用户体验，可以吸引用户的眼球。以前的网站，仅为了应付搜索引擎的搜索，对用户体验不是很重视。随着互联网的发展，重中之重是用户体验，许多网站都是把用户放在第一位考虑的，留住用户才能留住资本。所以，能留住用户首先需要网站美观，其次才是网站的功能。如果一个网站连基本的美观都做不到，那么第一眼已经失败了，因为现代社会生活节奏很快，用户没有耐心体验一个不美观的网站的功能，设计一个美观的网站才能吸引潜在的用户在网站上完成成交。

1.3.2 交互性

前端开发有几样任务是需要完成的，比如动画特效，复杂的交互功能，UI设计，可拓展性和后期维护等。

在前端开发过程中，有几项任务是必须完成的。例如U设计，动画特效，复杂的交互功能和后期维护等。

网站交互设计的目的在于增加网站界面的友好性，提高可用性，从而使用户能够方便快速和有效地网站特定的功能和用户自己想要完成的功能。

交互设计包含的内容：

1. 界面设计：界面设计不仅仅是UI方面的设计，还涉及到此网站为用户能提供何种功能，用户可以完成何种操作。
2. 导航设计：导航设计是一个网站设计尤为重要的地方。因为用户不是编写代码人员。没有简单易懂的样式提示用户如何操作，那么你这个网站便是失败的产品。
3. 信息设计：信息设计就是在用户使用网站的过程中，会有一些温馨或必要的提示，避免用户误操作或不会操作。

1.3.3 性能

网站的性能，也是决定用户是否在此网站停留的重要原因。用户浏览一个网站，最花费的时间是前端花费的时间，而不是后台应用程序处理以及数据库等消耗的时间。根据web优化的黄金法则:前端程序大部分时间则花在各种页面元素，如图像、样式表和脚本等的下载上，因此这占用了80%的最终用户响应时间。因此在编写代码的时候，减少HTTP请求次数，这是解决页面快速显示的主要途径之一。

1.4 本章小结

本章对顺丰优选网站开发的研究背景和意义进行了阐述。并阐述了国内外类似网站的情况，且点出了课题研究的目的和方式。最后指出课题的研究内容和拟解决的问题，只要是前端布局设计、交互性和性能这几个方面去考虑。让我们对项目有一个大致的了解和熟悉。

第2章 方案论证

2.1 相关技术

Web前端开发人员需要掌握的基础要素有三个:HTML，CSS和JavaScript,这就要求前端开发人员不仅仅要掌握基本的Web前端开发技术、网站性能优化、SEO和服务器端的基本知识,还要掌握基本的页面设计技巧。而且还要学会运用各种开发工具进行辅助开发以及理论层次的知识，包括浏览器分层支持、分层语义模板、代码的可维护性和组件的易用性等。

2.1.1 HTML5

超文本标记语言，用来搭建网页的DOM结构，是Web前端的缺一不可的基础技术之一。

HTML5是HTML最新的修订版本，2014年10月由万维网联盟（W3C）完成标准制定。使用HTML5技术可以使网页端浏览器减少对于AdobeFlash等插件的需求，同时还可以提供更多的网络应用标准集。

HTML5不仅增加了很多全新的语法特征，还定义了处理各种非法文档的细节。这就使得各种不同浏览器、客户端可以统一一致地处理各种可能出现的语法错误。

从用户的角度来说，网站使用HTML5可以明显提高用户体验。在HTML5中由于使用了CSS3特效样式、Canvas、webgl等技术，不仅加强了网页的视觉效果，甚至能够使用户在网页当中看到三维立体特效。

从开发者角度来说，拥有跨平台、多终端适配等特点的HTML5技术，可以大力减少针对不同终端、不同系统适配的成本。

HTML5添加了一些新特性：

1. 用于绘画的canvas元素
2. 用户媒介回放video和audio元素
3. 对本地离线存储的更好支持
4. 新的特殊内容元素，比如article、footer、header、nav、section
5. 新的表单空间，比如calendar、date、time、email、url、search
6. CSS3

层叠样式表，用来控制网页元素的显示方式，是Web前端缺一不可的基础技术之一。CSS3是最新的CSS标准，增加了一些新特性：

1. 新的选择器:如：nth-child(n)，last-child, nth-last-child(n)等
2. RGBA和透明度，如 rgba (123,155,20,.5);或opacity：0.5
3. 多栏布局，相关属性：column-count, column-gap, column-rule等
4. 多背景图，允许给一个元素添加多层背景图片
5. Word wrap, word-wrap属性来防止太长的字符串溢出
6. 阴影，相关属性：text-shadow, box-shadow
7. @font-face属性，声明自定义字体
8. 边框图片和圆角，相关属性:border-image, border-radius
9. 盒子大小，box-sizing属性使你可以指定浏览器如何计算一个元素的height和width值。
10. 媒体查询，让你为不同的设备基于他们的能力定义不同的样式
11. 语音，可以让你为屏幕阅读者指定语音样式，相关属性：voice-volume, voice-balance , speak , voice-family, voice-rate, voice-stress等

2.1.2 SSM

2.1.3 JavaScript/DOM/Ajax/JSON

JavaScript是轻量级的编程语言，可用于HTML和web，也可以用在用户服务器、PC、笔记本电脑、平板电脑和智能手机等设备。jQuery是一个JavaScript库，极大地简化了JavaScript的编程。其最大的优点是代码简洁干练，语义简单易懂。JQuery代码极其小巧，这是其它的JS库所不及的。同时，它兼容CSS3，并且是跨浏览器的。

jQuery还有一个比较大的优势是，它可以将JS代码与HTML代码完全分离。这样对于代码的维护、修改是极其方便的。

并且，JQuery的插件数量丰富，使用简单，开发者可以轻易通过插件实现更多的功能。

DOM，即文档对象模型，在页面加载时创建，通过JavaScript可以操作DOM，对DOM树进行增删改查操作。

Ajax是一种页面局部加载技术，无需加载整个页面，这是改善用户体验方面绝对重要的一种技术。

JSON，是JavaScript的对象表示法，也是一种轻量级的文本数据交换格式，其独立与语言，具有自我描述性，易于理解，传输速度快。

2.1.4 网站重构

构建一个灵活的前端MVC框架，即以HTML作为信息模型(Model)，CSS控制样式作为视图显示(View)，JavaScript负责调度数据和实现某种展现逻辑(Controller)，这是网站重构的本质。同时，我们要进行高效率，高质量开发以及协作开发，就需要代码具有很好的复用性和可维护性。通过网站的重构，可以让前端代码组织更有序，还可以显著改善网站的性能，还能提高代码的可维护性，并且让网站对搜索引擎也更友好，还可以为网站带来更好的用户体验，是的下载文件更小，页面显示速度更快。

2.1.5 Web优化

一个好的网站的美观的页面设计和完善的服务功能是必要的，但求性能能够满足用户的需求也非常重要，比如在使用网站时，用户看中了这个商品，他首先需要查看商品详情，才能决定自己是否购买，如果页面延迟时间太长，那么用户很可能会失去对这个商品的兴趣，则不会购买，久而久之用户会放弃使用这个网站。所以我们需要进行前端优化。

前端优化的途径有：

1、 减少Http请求

2、 减少文件大小

3、 将样式表放在顶部

4、 将script放在底部

5、 减少DNS查询

6、 避免重定向

7、 杜绝http 404错误

8、 避免使用CSS表达式

9、 使用外部javascript和CSS

10、删除重复脚本

11、使用内容发布网络（CDN的使用）

12、添加Expires头

13、配置ETag

14、使Ajax可缓存

2.1.6 Web 布局

1、固定布局（Fixed Layout）： 固定布局即在设置网页元素的宽高的时候，使用的固定的值[3]。所以，无论访问者屏幕的分辨率是多少，或者网页的窗口大小发生改变，网页都显示相的宽高。

固定布局优点：固定宽度的布局是最简便、最常用的布局方法；它在所有

浏览器都支持且表现一致，所以不会受到图片、表单、视频和其他固定宽度内容的麻烦；而且知道了每个元素的精确宽高，就能够对它们进行精确的布局。

固定布局缺点：因为固定不变，没法适应灵活多变的布局；无法充分利用

可用的空间；若屏幕分辨率变化，则会出现难看的滚动条；

2、流体布局：

流体布局，也被成为流式布局，尺寸是用百分数而不是像素设置的，因此

可以自适应用户的分辨率[3]。

流体布局优点：能够相对于浏览器的窗口进行伸缩；能够非常高效的使用

空间；设计更为美观，用户体验更好。

流体布局缺点：在窗口宽度较小时，行变得非常窄，很难阅读，在多列布局中也是如此，因此有必要添加加以像素或em为单位的min-width，从而防止布 局变得太窄，但min-width设置得太大，也会出现难看的滚动条。

3、弹性布局

弹性布局是相对于字号，而不是浏览器的宽度来设置元素的宽度的。以

em为单位设置宽度，可以确保在字号增加时整个布局随之扩大。

弹性布局优点：恰当的弹性布局对用户十分友好，它是固定布局和流动布

局的结合，堪称完美。比较受设计师们喜欢。页面中所有元素可以随着用户的

偏好缩放可以将行长保持在可阅读的范围之内，这对视力弱或有认知障碍的人

有其有用。

弹性布局缺点：弹性布局的一些问题与固定宽度布局相同，比如不能充分的利用可用空间；当字号增加时，整个布局也会加大，便会出现难看的滚动条；理解起来比较困难，设计开发时需要花费更多的时间，效率低下。

2.2 可行性分析

2.2.1 技术可行性

本网站将采用html5，CSS3,JavaScript，jQuery等前端技术和框架，结合subline text ，Chrome，eclipse, mysql等工具进行测试开发。

目前，在谷歌等国际巨头的推动下，HTML5技术飞速发展，各类游戏、视频网站均逐步转换为使用HTML5开发，这就给了我们技术层面上的保证。

同时，HTML5跨平台、多终端适配等特点，更是使得HTML5成为未来网站开发的必需工具。

2.2.2 操作可行性

顺丰优选平台是一个为客户提供最简洁的购买全球优质美食的平台,以“优选商品,服务到家”为宗旨,依托线上电商平台与线下社区门店,为用户提供日常所需的寰球优质美食。整个网站具备完整独立的选购功能，网站结构简洁明了，用户容易使用，操作简单。使用者并不需要拥有太多的计算机知识，也无需处理复杂使用逻辑，即可使用。

2.2.3 经济可行性

目前来看，虽然网站的前期开发推广对初期的投资需求较大，但是，如果本网站成功上线，将会因为其全球优质美食购买平台的定位，吸引大量用户，在激烈的市场竞争中脱颖而出。因此，从长远的眼光来看，一旦网站上线，其带来的收益将会远远高于投入。

2.3 结论

2.4 本章小结

本章主要介绍了网站设计开发过程中需要的一些基础技术，基础技术使用的已经很频繁了，所以只是略作介绍。因此课题以PC端开发为主要内容，本章则对相应的技术背景进行了介绍，以便更好的理解网站设计开发的思路。

第 3 章 需求分析

3.1 需求分析概述

需求分析对于系统的开发说件很重要的工作，是软件开发过程中尤为重要的环节，完善的需求分析是软件开发的一个基础。因为用户是第一位的，所以需求分析需要研究用户究竟喜欢什么，需要什么，这样才能做出一个美观且实用的网站。以下对顺丰优选网站前端的需求进行了详细的分析。

3.2 数据流分析

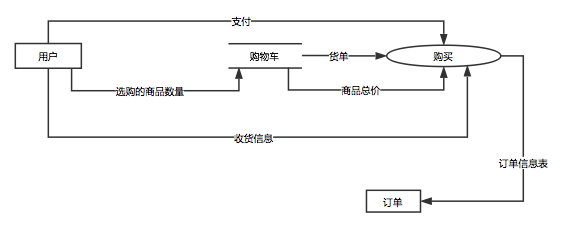
 图3-1 顺丰优选购物系统顶层图

图3-2 顺丰优选购物系统0层图

本系统的数据主要为购买商品服务，故数据的流通本质是促使用户买到自己心仪的商品。当用户选购商品后，商品数据流向购物车，用户填写购物信息后，购买商品，支付总价，生成订单，用户即可在“我的订单”中查看自己所买的商品信息。

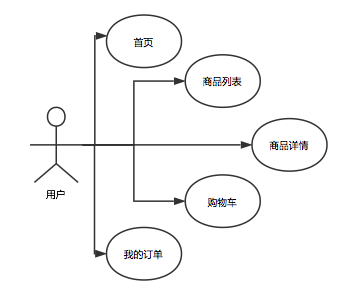
3.3 用例分析

图3-3 系统用例图

本系统为网上购物系统，主要实现的功能即为购买商品功能。

3.4 功能分析

作为顺丰优选线上网站，用户体验是第一要务。所以在做本系统的时候用户体验是我最先想到的。用户可以在本网站完成对海外一些商品的购买服务。

1. 提供友好的首页页面，展示商品情况。Banner轮播图展示，楼层导航方便用户操作。
2. 商品列表页面，瀑布流展示商品分布。
3. 商品详情页面，使用放大镜效果更好的展示商品的效果，介绍商品具体情况，加入购物车。
4. 购物车页面，对加入购物车的商品进行增加，删除，修改。
5. 确认订单页面，填写基本信息，方便收货。
6. 提交订单页面，提交订单，完成支付。
7. 查看订单页面，查看订单状态，对订单状态进行管理。
8. 收货地址页面，管理收货地址，方便快捷地购物。

3.4.1 系统包图

3.4.2 系统类图

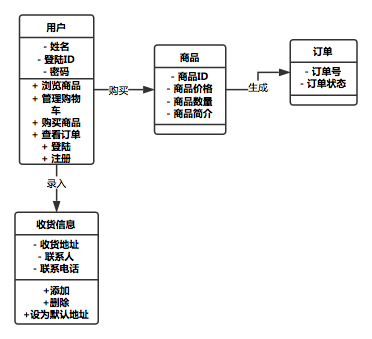


图3-5 顺丰优选购物系统类图

3.5 本章小结

本章主要为顺丰优选购物系统的需求分析，经过数据流分析，功能分析，画出了系统包图和系统类图，对于这个系统的功能有了更进一步的规划。

第4章 过程论述

4.1 概要设计

4.1.1 前端总体开发流程及设计

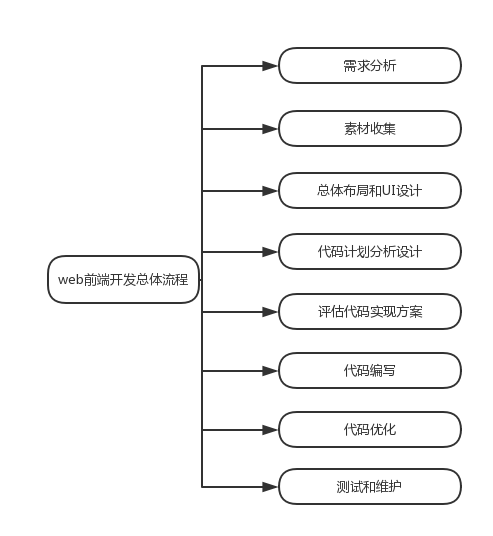
Web前端虽然入门极其简单，但是也有其复杂的一面，前端技术及知识面博大精深。在前端开发之前，必要的准备和设计工作是必须的。首先我们设计我们的各种UI界面的布局和排版，而且前期的素材收集工作也是必要的，不然等到用到时再去找素材会降低效率。设计完成了之后，也并不是立即就开始写代码，这样没有计划的写起来，最后只会让自己的代码十分难看，虽然有测试维护的工作，但是这也会浪费很大的时间。所以，一个有计划，有目的的进行的前端开发工作是非常有必要的。前端开发的总体流程如下图所示:

图4-1 前端开发总体流程图

4.1.2 组织结构设计

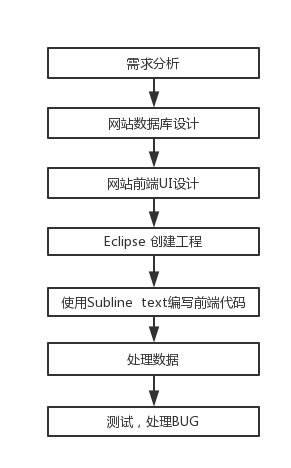
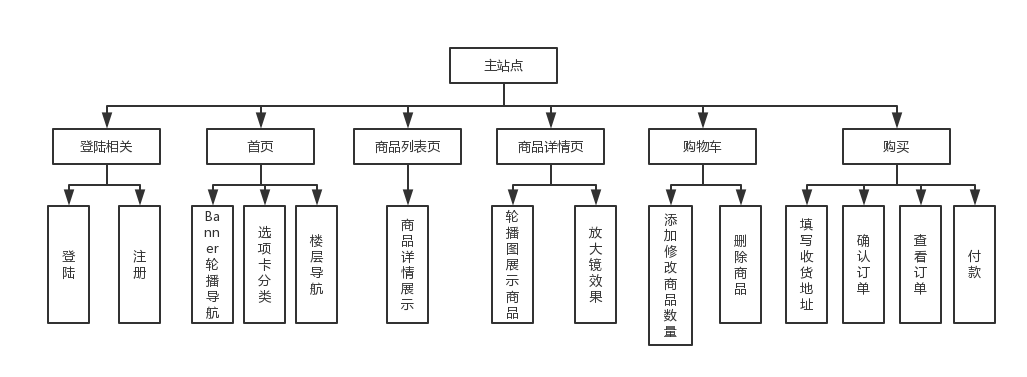
本网站采用的开发模式在目前的软件设计应用中较为流行，将软件应用系统主要客户端做成没有任何业务逻辑和业务数据的纯展示层，最大限度的提高了系统的安全性和稳定性的保护。本系统的设计流程如图：

图4-2 顺丰优选购物平台设计流程图

4.1.3 前端总体布局和UI设计

● 功能模块分布

前端开发的内容主要由这些模块组成:登录项关、首页、商品列表、商品详情、购物车、确认订单组成。如下图所示:

图4-3 顺丰优选购物系统功能模块分布图

● CSS 元素

CSS是控制页面样式的重要元素，而对于选择页面上的元素，CSS有多种选 择器，基础的选择器有id选择器，类选择器，如：#index,.nav\_top；此外还有标选择器，属性选择器，伪类选择器等等，但是，如果页面上元素的层级比较多的话，那么就会出现选择器很长，通常这会使得CSS渲染页面的时候出现性能问题。比如：

#index div .className p:nth-child(2){

color:#1155ce;

}

由于选择器的渲染顺序是：选择符从右向左进行匹配的，只要当前选择符的左边还有其他选择符，样式系统便会继续向左移动，直到找到与规则匹配的元素为止。所以，选择器多了并不一定是好事，虽然便于你理解选择器的意思，但是，我们还有一种更为简便的方式处理这种问题，可以让选择器执行起来不仅效率高，而且便于理解代码，那就是利用CSS选 择器中的优先级来处理，也即CSS的层叠性和特殊性。

在写层叠样式表的时候，经常会出现样式污染，也即多个样式寻找同一元 素的，从而导致样式冲突和样式的不确定性，CSS 每个规则分配一个重要度， 重要度层次如下：

1、 标准！important 的用户样式

2、 标准！important 的作者样式

3、 作者样式

4、 用户样式

5、 用户代理和浏览器应用的样式

CSS样式规则的次序是由选择器的特殊性决定的，若是特殊性相同，后面定义的规则覆盖先定义的规则。

规定给每种选择器被分配一个数字值，求出规则中的的每个选择器的值的和，由此计算出规则的特殊性。选择器的特殊性可以分为四个等级：

1、行内样式，数量一般为 1

2、ID 选择器的数量

3、类、伪类和属性选择器的数量

4、类型选择器和伪元素选择器数量

此外，CSS 选择器的执行效率从高到低有一个排序[4]：

1、id 选择器（#myid）

2、类选择器（.myclassname）

3、标签选择器（div,h1,p）

4、相邻选择器（h1+p）

5、子选择器（ul < li）

6、后代选择器（li a）

7、通配符选择器（\*）

8、属性选择器（a[rel="external"]）

9、伪类选择器（a:hover, li:nth-child）

根据 CSS 选择器的匹配原则和执行原则我们总结如下：

1. 尽量用单层选择器代替后代选择器和伪类
2. 写样式尽量用类选择器，虽然 ID 选择器效率最高，但 ID 选择器一般 是留给 JavaScript 使用。
3. 避免使用通用的选择器：.content \*{color:red;}
4. ID选择器尽量单独使用： #timemenu{…}
5. 避免使用标签限制 class 选择器：div.className {…}
6. 避免使用子选择器：ul>li
7. 使用继承

4.1.4 数据库设计

本系统是电商购物平台，所以最主要的表为以下四张表：

表4-1 用户表

表4-2 商品表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | 字段数 | 是否为主键 | 是否为空 | **注释** |
| ProductId | Char | 10 | PK | Not Null | 商品ID(系统自增加) |
| ProductName | Char | 20 |  | Not Null | 商品名称 |
| Price | Char | 10 |  | Not Null | 单价 |
| Imgurl | String | 20 |  | Not Null | 图片地址 |
| Desc | String | 50 |  |  | 描述信息 |
| Comment | String | 50 |  |  | 评价 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | 字段数 | 是否为主键 | 是否为空 | **注释** |
| Username | Char | 5 | PK | Not Null | 用户名 |
| UserID | Int | 10 |  | Not Null | 用户ID(系统自增加) |
| Password | Varchar | 20 |  | Not Null | 密码 |
| Address | Char | 30 |  | Not Null | 收货地址 |
| PhoneNumber | Char | 11 |  | Not Null | 联系电话 |

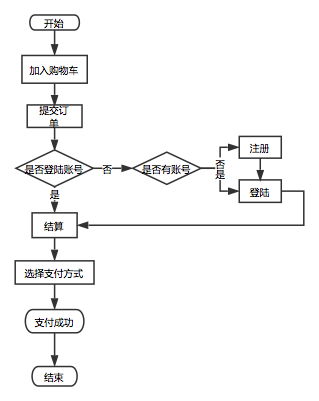
表4-3 订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | 字段数 | 是否为主键 | 是否为空 | **注释** |
| Id | Char | 10 | PK | Not Null | 订单号(系统自增加) |
| Status | String | 20 |  |  | 订单状态 |
| UserID | Char | 10 |  | Not Null | 用户ID(系统自增加) |
| ProductID | Char | 10 |  | Not Null | 商品ID(系统自增加) |
| OrderDate | Varchar | 30 |  | Not Null | 订单时间 |
| Name | Char | 5 |  | Not Null | 收货人姓名 |

表4-4 收货地址表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | 字段数 | 是否为主键 | 是否为空 | **注释** |
| PhoneNumber | Char | 11 | PK | Not Null | 联系电话 |
| Name | Char | 5 |  | Not Null | 收货人姓名 |
| Address | Char | 30 |  | Not Null | 收货人地址 |

4.1.5 程序流程图

图4-4 顺丰优选购物系统程序流程图

如图所示为本系统的程序流程图，本系统为方便用户购物，支持在未登录时挑选商品，且可以加入购物车，提交订单时会检测是否登陆账号，若未登录则需登录，未注册的用户需要注册，这时会弹出页面，但登录后购物车内容不会消失，这样一来就优化了用户的购物体验。

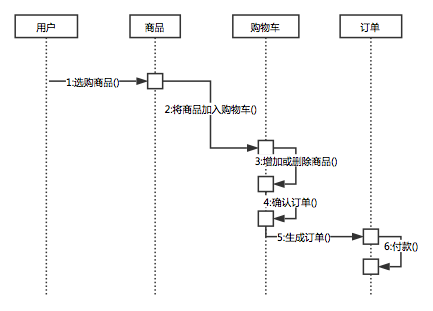
4.1.6 时序图

图4-5 顺丰优选购物系统时序图

4.1.7 活动图

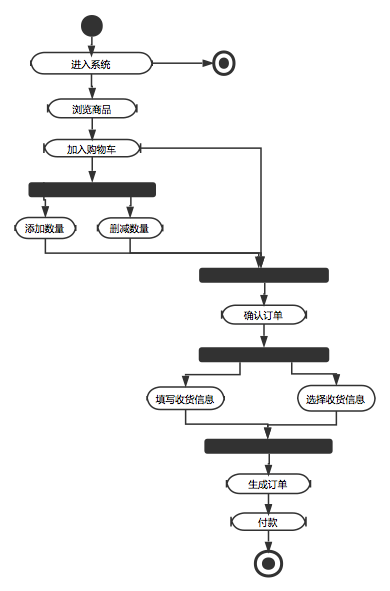


图4-6 顺丰优选购物系统活动图

4.2 详细设计

4.2.1 首页banner轮播图

使用jQuery实现banner的无缝滚动,用户可根据下方小圆点的指示看到当前小圆点的图片,用户没有操作时，图片将自动轮播。

图4-7 首页轮播图

核心代码如下：

//大图轮播

//获取初始值，轮播图的名字，图片的高度、长度

var $topics = $(".topic"),

imgWidth = $(".topic").width(),

len = $topics.length,

nextIndex = 1,

time = null;

//轮播盒子的总长度

$(".b-list").width(imgWidth\*len);

//轮播

time = setInterval(move,3000);

//动态添加小圆点

$.each($(".topic"),function(index,element){

var Index = index +1;

$("#pages").append("<div>"+Index+"</div>");

$("#pages :first-child").addClass("curr");

$("#pages :last-child").data("index",index);

});

// 鼠标移入运动

$("#pages div").on("mouseenter",function(){

clearInterval(time);

nextIndex = $(this).data("index");

move();

time = setInterval(move,3000);

});

//运动

function move(){

var \_left = -nextIndex \* imgWidth;

//小圆点改变样式

$.each($("#pages div"),function(index,element){

$("#pages div").eq(index).removeClass("curr");

});

$("#pages div").eq(nextIndex).addClass("curr");

nextIndex++;

if(nextIndex>=len){

nextIndex=0;

}

$(".b-list").animate({"left":\_left});

}

4.2.2 首页楼层导航

通过判断用户滚动页面的距离来显示楼层导航,当用户滚动到当前楼层时，楼层样式改变。用户可以点击楼层跳转到当前楼层。逻辑js如下：

//楼层导航

var imgoneHeight = $(".img-one").offset().top,

winHeight = $(window).height();

//获取最后一层的高度

var lastHeight = $(".drink").offset().top;

//开关变量

var isMoving = false;

//当页面滚动的一定距离是显示楼层

$(window).on("scroll",function(index,element){

if(!isMoving){

//获取当前页面的高度

var srcTop = $(this).scrollTop();

//通过两个值来确定楼层导航出现和消失

if(srcTop > imgoneHeight - winHeight&&srcTop<lastHeight){

$(".floor").show();

}else{

$(".floor").hide();

}

$(".friut").each(function(index,element){

var \_top = $(this).offset().top;

//页面滚动到当前楼层时，改变样式;其他兄弟楼层为其他样式

if(srcTop > \_top - winHeight/2){

$(".floor>li").eq(index).addClass("caur").children("span").

show().end().siblings().removeClass("caur").children("span").hide();

}

});

}

});

//鼠标移至其他楼层是显示本层楼的汉语简介，并且其他楼层样式的消失

$(".floor li").hover(function(){

$(this).children("span").show();

},function(){

$(this).not(".caur").children("span").hide();

}).on("click",function(){//点击楼层滑动至所点楼层

isMoving = true;

var index = $(this).index();

var \_top = $(".friut").eq(index).offset().top;

$(this).addClass("caur").children("span").show().end().siblings().

children("span").hide();

//运动过程

$("html,body").animate({"scrollTop":\_top},1000,function(){

isMoving = false;

});

});

4.2.3 商品展示页面

在商品展示页面用户可以看到某个分类的商品列表，能够在自己想要寻找的分类中找到自己心仪的商品。

比如牛肉类产品的分类，此页面显示了所有牛肉分类的产品，在此页面可将物品加入购物车，加入购物车时 可以选择商品的数量。点击某商品进入商品详情页面。

当用户鼠标移入某个商品时，此商品会显示背景阴影效果。更好的突出商品。这里使用css3的box-shadow属性显示阴影效果。

.dmrt-product:hover{

border:none;

box-shadow:2px 2px 2px 2px #969696;

}

4.2.4 商品详情页面

此页面展示了商品的详细数据，包括商品的价格、产地、详细信息介绍等等。

● 轮播图片介绍商品

如下图所示，通过图片轮播的方式来介绍商品的详细信息。用户可以选择昨天的小图来切换中间的图片显示，当鼠标移动到中间图片时。右方显示此图片的放大效果。即完成放大镜的效果。



图4-8 放大镜效果图

此效果使用jQuery完成，关键代码如下：

/\*\*\*\*\*\*\*\*\*\*\*放大镜效果\*\*\*\*\*\*\*\*\*\*\*/

// 小图轮播

var nextIndex = 1,

$imgs = $(".op-box img"),

imgHeight = $(".op-box img").height(),

len = $imgs.length;

$(".op-box").height(imgHeight\*len );

//上一张

$(".op-prev").on("click",function(){

move();

});

// 下一张

$(".op-next").on("click",function(){

move();

});

//移动函数

function move(){

var \_top = -nextIndex \* imgHeight;

nextIndex++;

if(nextIndex>len-5){

nextIndex =0

}

$(".op-box").animate({

"top":\_top,

});

}

//放大镜

//鼠标移到小图切换大图

$.each($(".op-box img"),function(index,element){

$(this).on("mousemove",function(){

$(".op-middle img").attr({

src:"img/middle\_1800240143\_1\_"+(index+1)+".jpg",

});

$(".op-big img").attr({

src:"img/original\_1800240143\_1\_"+(index+1)+".jpg",

});

});

});

var middleWidth = $(".op-middle").width(),

middleHeight = $(".op-middle").height(),

popWidth = $(".op-pop").width(),

popHeight = $(".op-pop").height();

$(".op-middle").hover(function(){

$(".op-pop,.op-big").fadeIn();

},function(){

$(".op-pop,.op-big").fadeOut();

}).on("mousemove",function(event){

//计算出遮罩在整个文档中的绝对定位(即光标在遮罩中心)

var \_left = event.pageX - popWidth/2,

\_top = event.pageY - popHeight/2;

$(".op-pop").offset({

"top":\_top,

"left":\_left

});

//获取到遮罩相对.op-middle的定位

\_left = $(".op-pop").position().left;

\_top = $(".op-pop").position().top;

console.log(\_left,\_top);

//越界

if(\_left<0){

\_left = 0;

}else if(\_left > middleWidth - popWidth){

\_left = middleWidth - popWidth;

}

if(\_top<0){

\_top = 0;

}else if(\_top >middleHeight - popHeight){

\_top = middleHeight -popHeight;

}

//重新定位

$(".op-pop").css({

"top":\_top,

"left":\_left

});

//放大镜位置

$(".op-big img").css({

"top": -2\*\_top,

"left": -2\*\_left

});

});

● 加入购物车、商品详情介绍

当用户喜欢此商品时，可以将此商品加入购物车。在加入购物车的时候可以选择自己所需要的数量。

当用户处于登录状态，加入购物车后用户可以选择继续购物或者查看自己的购物车。如果用户点击查看购物车时，则跳转至购物车页面。

当用户处于未登录状态，页面则跳转至登录页面。

下方则是商品的详情介绍。这里我是使用顺丰优选原网站的图片，加它的图片下载下来之后，拼接成商品的详情介绍。在商品详情介绍的旁边是已经购买此商品的部分用户的评价信息。以及顺丰优选网站对此商品作出的一些承诺。用户可以参看已经购买此商品的用户的一些评价来帮助自己更好的完成一次自己心仪的网上购物行动。

4.2.5 购物车页面

此页面只有当用户处于登录状态才可以看到，这里显示了用户加入购物车的商品简介以及用户所需支付的总金额。在此页面用户可以对单个商品进行数量上的加减操作，可以自己手动输入数量。也可以删除某个自己不想要的商品。同时可以删除所有商品。商品的总金额是随着商品数量的改变而进行改变的。

● 商品数量的改变

在此用户可以增加商品的数量，减少商品的数量以及手动填写自己需要的商品数量。核心代码如下：

// 加数量

$(".addnum").click(function(){

var amount = $(this).prev(".num").val();

amount++;

update(amount,$(this).parents(".cart\_body"));

});

// 减数量

$(".minus").click(function(){

var amount = $(this).next(".num").val();

amount--;

if(amount<1){

return;

}

update(amount,$(this).parents(".cart\_body"));

});

// 填数量

$(".num").on("blur",function(){

var amount = $(this).val();

if(/^[1-9]\d\*$/.test(amount)){

update(amount,$(this).parents(".cart\_body"));

}else{

alert("数量应该为数字");

var prod = $(this).parents(".cart\_body").find("a").data("product");

$(".num").val(prod.amount);

}

})

● 商品的操作

用户可以对加入购物车的商品进行修改。可以删除单个商品或者删除所有的商品。用户所有的操作完成之后。商品的总金额都会自动变化。这里我使用jQuery提供的方法快速的选择到所对应的节点进行应有的操作。

这里自己先编写了一个删除的总函数，后面所有的删除操作都调用一次这个方法，这样减少了重复代码，提高了自己的编写效率。核心代码如下：

// 删除选中

$("#del\_checked").on("click",function(){

$("#cart .ck:checked").parents(".cart\_body").find(".detele").click();

if(products.length==0){

$('.zhezhao').show();

$(".cart\_null").show();

}

});

当用户删除选中商品时，这样这个商品对应的input的值为checked，这样我便可以通过下面这段代码完成删除操作。

$("#cart .ck:checked").parents(".cart\_body").find(".detele").click();

在商品数量变化的同时，商品的总金额也会随之改变。在这里我写了一个计算商品总金额的函数calcTotal()用户计算商品的总金额。然后在每次改变操作之后调用这个函数，那么商品的总金额就会随着用户的操作改变而进行改变。核心代码如下：

function calcTotal(){

var sum = 0;

$("#cart .ck:checked").each(function(index,element){

var a = parseFloat($(this).parents(".cart\_body").

find(".sub").text());

sum+=a;

})

$("#subAll").text(sum.toFixed(2));

}

● 确认订单

对于所有的电商网站来说，当用户把自己所喜爱的商品加入购物车后。在付款之前会有一个填写收货地址的页面。同样本系统也提供这样的功能。用户可以通过点击继续购物返回主页继续选购商品，也可以点击去付款进入填确认订单页面。

4.2.6 购买商品

对于电商网站来说，确认订单页面可以说是非常重要。这里可以说是商家如何将用户喜欢的商品迅速的送到用户手中的关键。用户填写完自己的收货地址，付款之后便可以在家等待自己喜欢的商品送货上门了。

● 省级市联

省级市联是前端经常出现的一个功能。用户可以选择自己所在的具体地址。这里我用的是省、市、县(区)三级联动。对于用户来说，当用户选择省份为吉林省时，那么下一个选择框里便是吉林省下属的所有市，第三个选择框里面便是所选市下属的所有县或者区。

为了实现这个功能，首先需要全国的省、市、县的json文件。然后通过遍历这个json文件，将json对象转换为数组对象。然后在对这个数组对象进行操作。为了完成这个功能我编写了一下三个函数:initProvince(),initCity(),initDistrist();

● 获取所有的地址信息

var address = [];

$.getJSON("js/address.json",function(data){

var provinces = data.regions;

$.each(provinces,function(index,province){

address[province.name] = [];

var cities = province.regions;

$.each(cities,function(index,city){

address[province.name][city.name] = city.regions;

});

});

initProvince();

});

● initProvince函数(省份初始化)

此函数获取所有的省份城市信息并将之填入所属的选择框中。核心代码如下：

// 省份初始化

function initProvince(){

$("#province").empty().append("<option value='-1'>请选择省份 </option>");

for(var provinceName in address){

$("<option value='"+provinceName+"'>"+provinceName+"</option>")

appendTo("#province");

}

}

● initCity函数(市级信息初始化)

此函数通过用户所选择的省份信息，显示此省份下属的所有城市信息。核心代码如下：

// 城市加载

function initCity(){

var provinceName = $("#province").val();

// 获取该省份下的所有城市

var cities = address[provinceName];

// 显示城市

$("#city").empty().append("<option value='-1'>请选择城市</option>");

for(var cityName in cities){

$("<option value='"+cityName+"'>"+cityName+"</option>")

.appendTo("#city");

}

$(".sheng").text(provinceName);

text\_sheng = $(".sheng").text();

};

● initDistrict函数(县区信息初始化)

此函数通过用户所选择的市级信息，显示此市级下属的个区县信息。核心代码如下:

function initDistrict(){

// 获取已选省份与城市

var provinceName = $("#province").val(),

cityName = $("#city").val();

// 获取选定城市下的所有区县信息

var districts = address[provinceName][cityName];

// 显示区县

$("#district").empty().append("<option value='-1'>请选择区县</option>");

for(var i in districts){

$("<option value='"+districts[i].name+"'>"+

districts[i].name+"</option>").appendTo("#district");

}

$(".shi").text(cityName);

};

● 确认商品信息

在此页面用户最终确认自己所选择的商品信息。所有信息无误之后便可以进行付款完成此次网上购物操作。

* 付款

当用户填写完收货地址，确认自己的订单无误之后，就进入了付款页面。用户可以根据自己喜好或者自己持有的支付方式来选择如何支付，本系统为用户提供了多种支付方式。

图4-9 支付页面图

4.2.7 我的订单

无论用户是否支付，均会生成订单号，存入订单表中，用户可在我的订单中查看订单状态。用户未完成支付时，订单状态则为待付款，此时用户可以随意取消订单，或选择立即付款。当用户完成支付，订单状态会变为待发货，发货之后会变为已发货，且物流动态实时更新。

图4-10 订单状态图

4.2.8 管理收货地址

当用户购买商品时，用户需填写收货地址，此收货地址可以是任何人的地址，一个用户可能有多个收货地址和收货人，此时需要管理收货地址来对收货地址进行管理。在此界面，用户可以对自己的收货地址进行删除、新增和修改原收货地址，并且可以选择是否选为默认收货地址。

若用户将某地址选为默认收货地址，那么在下次购物时商家会将商品发往默认收货地址。一般用户都会选择自己最常用的地址作为默认收货地址，这样可以节省用户时间，提高用户体验。



图4-11 收货地址管理图

4.2.9 登陆注册

* 注册

注册支持帐户名，密码，验证码即可注册成功。帐户名是13位的手机号码，密码为6-20位的字符，安全程度分为弱、中、强，当用户的密码只有一种字符时，会提示用户密码强度为弱，需要加强。

验证码为进入页面后自动生成的随机四位验证码，输入正确的验证码即可注册。注册成功后跳转登录页面。



图4-12 注册界面

* 登陆

输入已注册的用户名和密码，并且验证码都过后则登录成功，进入首页。



图4-13 登陆界面

4.3 本章小结

本章主要展示了实现后的页面效果，以及对其中的一些功能进行了详细的描述，并贴出了实现的代码。

第5章 软件测试与结果分析

5.1 前端开发中常见Bug及修复

前端开发最常见的任务是处理由浏览器的兼容性，JavaScript和CSS版本更新引起的问题，这些也常常是前端开发者比较头疼的问题。有经验的前端开发者，处理这些问题会很快，而对于经验不足的开发者来说，就需要查阅大量的资料挥着花费大量的时间来研究。所以，在此依据网络上前辈们的经验，对前端开发中常见的Bug进行归纳和总结。

5.1.1 IE浏览器常见Bug

1. 浮动元素的双倍Margin的Bug:

IE6以及其以下版本的一个经典Bug就是浮动元素的双倍margin值。给某个元素设置了float属性并且同时在float同一方向设置了margin值就会触发这个Bug。

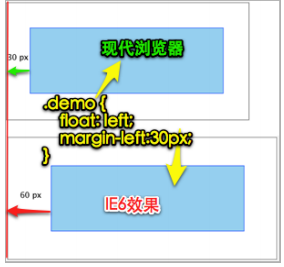


图5-1 BUG

解决方法:

#div{

Display:inline;

}

1. 设置元素的min-height和min-weight:

在前端开发过程中，我们经常需要控制元素的最小高度和最小宽度值，使得页面元素可以进行统一渲染，我们可以使用min-height和min-weight来控制，别的浏览器可以识别这两个属性，但是IE6却识别不了。如果必须要使用这个来达到统一的效果，针对IE6我们需要作出特别的处理，min-height的解决方法如下：

第一种方法事采用!important来解决,代码如下:

#demo{

min-height:200px;

height:auto !Important;

height:200px;

}

在IE6浏览器下的元素高度会根据自己内容的高度而定，内容高度低于min-height值，需要给元素一个显式的高度值，所以最后一句设置height值也设置为200px。

因为IE6不支持子选择器，所以还可以使用子选择器来修复，如下：

#demo{

min-height:200px;

height:200px;

}

html>body #demo{

height:auto; /\*只有现代浏览器才能识别\*/

}

1. 克服盒模型的Bug

在同时给某个元素设置了height和width的时候还设置了元素的padding值或border值就会触发盒模型Bug，元素的真正大小将会发生变化。如图5-2:

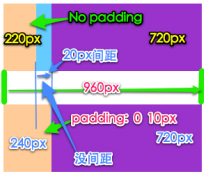


图 5-2 盒模型Bug

这个Bug在所有浏览器都会存在，解决方式是在div内部增加一个div,并且把这个新div的padding值设置为需要的值。

<div id="wrap">

<div id="left">

<div>left</div>

</div>

<div id="content">main</div>

</div>

CSS增加如下代码

#left div {

padding: 0 10px;

}

1. 块元素水平居中

这个Bug只会存在于IE6浏览器的怪异模式下，IE6在这个模式下无法识别margin属性的auto值，所以最直接解决的办法是在页面头部加上DOCTYPE或者在居中元素的父元素中加上“text-align:center”,然后在居中元素中加上“text-align:left”重新让元素文本左对齐来解决。

1. 列表li的楼梯Bug

在li中放置了一些元素内容并对其设置浮动，且li标签本身不浮动，就会在IE下触发li标签的楼梯Bug，如图5-3：



图5-3 li标签楼梯Bug

解决办法之一是给li标签也加上一个浮动，第二个解决办法是给li标签设置:“display:none”。

1. li标签空白间距:

使用列表是,IE会无故增加li与li之间的垂直距离。如图5-4：



图5-4 li 标签空白间距

解决方法是给a标签显式的定义一个宽度或者高度来触发IE浏览器的拥有布局:

li a {

width:100px;

}

或者给a标签加上一个浮动并且清除浮动，也可以解决：

li a {

display:block;

float:left;

clear:left;

}

还可以使li标签变为行内元素显式:

li {display:inline;}

li a {display:block;}

1. IE6下无法设置元素的微高

比如我们设置一个div的高度为2px,会出现这样:

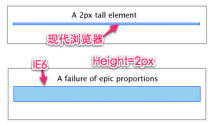


图5-5 IE6无法设置微高

在IE浏览器下，高度小于字号的设置会被拒绝，所以加上这两个属性和属性值即可解决:

font-size:0;

Line-height:0;

1. IE6下“躲猫猫”

出现这个Bug的条件是:浮动元素后面紧跟一些非浮动元素,并且浮动元素撑破了容器，并且非浮动元素中一些定义了:hover的链接，那么当链接在悬浮状态下就会在IE6及其以下版本触发这个奇怪而又无奈的Bug，想要解决这个bug，最好的方法便是清楚浮动。

1. overflow:auto与position:relative的碰撞

当两个块级元素都设置了overflow的值为auto或者hidden时,如果子元素设置position:relative并且其高度大于父元素时，就会产生子元素块不被隐藏会溢出父元素块，这只会在IE6和IE7中出现，而在IE8和FF还有IE5.5中又显示正常。

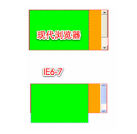


图 5-6 overflow和position碰撞

解决方法：

#right{

Position:relative;

}

1. 浮动层错位

当元素内容超出父容器定义的width时会出现浮动层错位的问题，在Firefox、IE7、IE8及其他浏览器中超出的内容仅仅只是超出边缘，这个很好解决。但是在IE6浏览器中容器宽度会错误地随内容宽度变大而变大，忽视定义的width值，并且在这个浮动元素之后若果还跟着另外一个浮动元素，那么浮动层错误bug变会出现了。

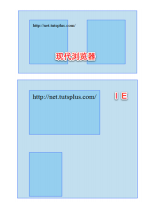


图 5-7 浮动层错位bug

解决方法：

#right{overflow:hidden;}

这样可以将超出的内容“切掉”。但是这样依旧会出现问题。虽然使用overflow:hidden;或overflow:scroll;可以将超出的内容“消失”,但是使用overflow:hidden会导致其他一些问题,如scroll会破坏设计。所以最好的解决方法便是使用固定布局或者是定义好宽度。

5.1.2 JavaScript兼容性Bug

1. 集合类对象问题:

Firebox下，只能使用[]获取集合类对象，IE下，可以使用()或[]获取集合类对象；可以统一使用[]获取集合类对象来解决[1]。

1. document.formName.item('itemName')问题:

IE下，可以使用 document.formName.item('itemName')或

document.formName.elements [“elementName”];

Firefox下，只能使用document.formName.elements[‘elementName’]。

解决办法是统一使用document.formName.elements[‘elementName’]。

1. 变量名与某HTML对象ID相同问题:

IE浏览器下HTML对象的ID可以作为document的下属对象变量名直接使用而Firefox不能使用[1]。但是在Firefox可以使用与HTML对象ID相同的变量名而IE不能用。

解决方法:

使用document.getElementById(“idName”) 代替 document.idName;

1. 自定义属性问题:

IE可以使用获取常规属性的方法来获取自定义属性[1]，，也可以使用

getAttribute()方法来获取自定义属性;但是在Firefox下，只能使用getAttribute() 获取自定义属性。

解决方法：

统一使用getAttribute() 获取自定义属性来解决。

1. input属性问题:

IE浏览器下 input.type 属性为只读;但是在Firefox下 input.type 属性为读写。

解决方法：

不修改 input.type属性;如果必须要修改这个属性的话,则可以先使得以前的input“消失”，然后在原来的位置插入一个新的input元素[1]。

1. window.event问题:

IE 中可以直接使用 window.event 对象，而 FF 中则不可以。解决方法如下:

第一种:

var theEvent = window.event || arguments.callee.caller.arguments[0];

第二种是将event 作为参数来传递:

function test(event){

var event = event || window.event;

}

1. event.x与event.y问题:

IE浏览器下，event对象有x、y属性，但是没有pageX、pageY属性；Firefox下，event对象有pageX、pageY属性，但是没有x、y属性。解决方法如下:

var myX = event.x?event.x:event.pageX;

var myY = event.y?event.y:event.pageY;

1. event.srcElement问题:

IE下，event对象有srcElement属性，但是没有target属性；Firefox下，event对象有target属性，但是没有srcElement属性[2]。解决方式如下:

srcObj = event.srcElement?element.srcElement:event.target;

1. window.location.href问题

在IE浏览器或者Firefox2.0.x浏览器下，可以使用window.location或window.location.href；

Firefox1.5.x下，只能使用window.location。

解决方法:

使用window.location来代替window.location.href。当然也可以考虑使用location.replace()方法。

1. 对象宽高赋值问题

Firefox中类似obj.style.height = imgObj.height的语句无效。解决方法:统一使用obj.style.height = imgObj.height + ‘px’;

5.1.3 CSS兼容性Bug

1. !important

随着IE7对!important的支持,!important方法现在需要对IE6做兼容。

例如:

#demo{

width:100px !important; /\*IE7+Firefox\*/

width:110px; /\*IE6\*/

}

1. CSS Hack方法

所有浏览器通用 height:100px;

IE6专用 \_height:100px;

IE7专用 \*+height:100px;

IE6、IE7共用 \*height:100px;

IE7、FF共用 height:100px !important;

1. 使用IE专用的条件注释

IE的if条件Hack

1. <!--[if !IE]><!--> 除IE外都可识别 <!--[endif]-->
2. <!--[if IE]> 所有的IE都可识别 <![endif]-->
3. <!--[if IE 5.0]> 只有IE5.0都可识别 <![endif]-->
4. <!--[if IE5]> 仅IE5.0与IE5.5可以识别 <![endif]-->
5. <!--[if gt IE5.0]> IE5.0以及IE5.0以上版本都可以识别 <![endif]-->
6. <!--[if IE6]> 仅IE6可识别 <![endif]-->
7. <!--[if lt IE6]> IE6以及IE6以下版本可以识别<![endif]-->
8. <!--[if gte IE6]> IE6以及IE6以上版本可以识别<![endif]-->
9. <!--[if IE7]> 仅IE7可识别<![endif]-->
10. <!--[if lt IE7]> IE7以及IE7以下版本可以识别<![endif]-->
11. <!--[if gte IE7]>IE7以及IE7以上版本可以识别<![endif]-->
12. FLOAT闭合 (clearing float)

网页元素的显示错位很多时候都是因为使用了float浮动而没有清除浮动。Div元素无法自适应高度也是如此[1]。在一个父元素下包含多个子元素时，如果父元素没有设置float而其子元素却设了float的话,父元素无法包住整个子元素，解决办法如下:

1. 给父元素也设上float属性
2. 在所有子元素后新加一个空元素，但这在有些浏览器中会产生空白间隙。
3. 万能float闭合

将以下代码加入公共CSS中，给需要闭合的div加上 class=”clearfix”即可:

<style>

/\*Clear Fix\*/

.clearfix:after{

content: ".";

display: block;

height: 0;

clear: both;

visibility: hidden;

}

.clearfix {

display: inline-block;

}

/\*Hide form IE Mac\*/

.clearfix{

display: block;

}

/\*End hide form IE Mac\*/

/\*end of clearfix\*/

1. 在父DIV的CSS中加上overflow:auto

5.2 前端的测试方法及测试工具的使用

5.2.1 本地测试方式

1. PC本地调试方式

静态页吗:直接下载页面到网络目录中，用相应的浏览器打开页面进行测试。

动态页面:动态页面则需要部署服务器，再将本地的Host指向接口所在域，将需要查看和调试的页面文件下载到网络目录中，然后在浏览器中输入http://本地 IP/test.html，页面将本地显示成功。

1. 移动端查看以简单测试方法
2. 将本地的Host指向接口所在域
3. 再将需要查看和调试的文件下载到网络目录中
4. 在PC端安装代理软件(fiddler)和页面二维码生成插件(chrome浏览器有二维码插件)
5. 在PC端安装node，使用npm安装weinre
6. 设置移动端网络代理指向PC端本地IP，端口8888
7. 打开fiddler和node，用移动端扫描PC端待检测页面生成的二维码
8. 查看页面，且可以在PC端实时查看响应的js，css，html文件

这个过程只要是使用fiddler来映射网络地址，是移动端的访问通过映射之后访问PC上网络目录里边的文件，这样就可以在移动端查看效果，在PC端来实时修改。

5.2.2 测试开发工具

1. 目前前端开发工程师们比较青睐的开发浏览器是Chrome,Chrome也以其强大的功能紧紧抓住了开发者的心。

Google Chrome一共提供了8大组工具[11]:

1. Elements：我们可以直接查看页面的DOM结构，且我们可以任意修改它来查看效果。
2. Network：可以让我们查看页面加载过程中请求了哪些资源，花费了多少时间和资源大小等。此外，还可以查看HTTP的请求头，返回内容等。
3. Sources：主要用来调试JS，也可以看到各种资源文件。
4. Timeline：可以查看浏览器在加载页面时所花费的时间并且用户分析报告，包括事件，资源下载、JavaScript处理、CSS样式计算等花费的时间都展示在Timeline中。
5. Profiles：可以分析查看页面和Web应用的执行时间以及内存使用情况。
6. Resources：对本地缓存(IndexedDB、Web SQL、Cookie、应用程序缓存、Web Storage)中的数据进行确认和编辑。
7. Audits：可以分析页面加载的过程，且提供提升响应速度、减少页面加载时间的方案。
8. Console：控制台，显示各种错误与警告信息，可以在其中编写shell命令用来和文档、开发者工具交互。

Chrome还可以很好的模拟手机环境，只需要设置一下开关就行。

1. Fiddler：Fiddler是一个http协议调试代理工具，使用它可以抓取你的电脑和互联网之间的http通讯。也可以设置断点，来查看所有的“进出”Fiddler的数据(指cookie,html,js,css等文件)。

5.3 测试用例和测试结果

5.3.1 测试用例摘要

部分测试用例如表5-1所示

表5-1 测试用例摘要表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **用例名称** | **摘要** | **前置条件** | **执行方式** | **操作步骤** | **预期结果** |
| 1 | 首页banner轮播导航 | Google Chrome浏览器 |  | 手动 | 用户鼠标移动图片下方编号时，banner轮播图滚动至编号对应的图片 | 打开页面时，进行无缝自动轮播。当用户鼠标移入下方某个编号时，图片滚动至编号对应的图片。 |
| 2 | 楼层导航 | Google Chrome浏览器 |  | 手动 | 用户滚动鼠标滑轮使页面向下发滚动。 | 当用户滚动至一定距离时出现楼层导航。当滚动至两个楼层交界处时，楼层自动切换。点击某个楼层，页面自动滑动至点击楼层所在位置 |
| 3 | 选项卡 | Google Chrome浏览器 |  | 手动 | 用户鼠标移入导航栏出现选项卡导航 | 用户鼠标移入banner轮播图上方的导航时出现选项卡导航。出现不同的内容进行切换选择。 |
| 4 | 放大镜效果 | Google Chrome浏览器 |  | 手动 | 用户鼠标移入商品详情页的图片时，右边会出现此图片的放大效果。 | 用户鼠标移入商品详情页的图片。出现放大效果。可以点击左边轮播导航的图片进行切换图片显示 |
| 5 | 购物车功能 | Google Chrome浏览器 |  | 手动 | 用户将喜爱的商品加入购物车 | 进行商品的增删改。用户可以添加、减少商品的数量。同时也可以删除商品。商品的总金额会随着这些操作的改变而实时改变。 |
| 6 | 省级市联 | Google Chrome浏览器 |  | 手动 | 用户填写自己的收货地址 | 用户填写自己的收货地址 |
| 7 | 随机生成验证码 | Google Chrome |  | 手动 | 登录，注册时用于验证 | 在登录注册时，系统随机生成验证码，提供安全性。 |

5.3.2 测试结果摘要

部分测试结果如表5-2所示:

表5-2 测试结果摘要表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bug\_id** | **标题** | **优先级** | **创建时间** | **解决时间** | **描述** | **Bug解决情况** |
| YY-001 | 首页导航图无法自动轮播 | 重要 | 2017-4-15 10:30:45 | 2017-3-15 12:58:20 | 首页的导航图不能自动轮播，影响用户体验 | 已解决 |
| YY-002 | 楼层导航不对等 | 重要 | 2017-4-15  14:10:57 | 2017-4-15  18:00:00 | 首页楼层导航时，当页面滚动到一定距离时，出现的楼层信息和需要的信息不对等。 | 已解决 |
| YY-003 | 商品列表也商品排列问题 | 重要 | 2017-4-16  11:30:57 | 2017-4-16  14:20:45 | 当用户鼠标移入某个商品时,此商品应该出现阴影。但是打乱页面布局 | 已解决 |
| YY-004 | 商品详细页面放大镜效果 | 重要 | 2017-4-17  9:10:20 | 2017-4-18  11:30:24 | 当用户鼠标移入需要放大的图片，左边显示放大后的效果。但是没有进行边缘设置。当鼠标移出页面会出现偏差。 | 已解决 |
| YY-005 | 介绍商品详细信息 | 中等 | 2017-4-18  16:30:25 | 2017-4-18  20:30:30 | 布局问题 | 已解决 |
| YY-006 | 购物车商品获取问题 | 重要 | 2017-4-20  10:30:25 | 2017-4-20  12:00:15 | 接口问题。获取数据出错 | 已解决 |
| YY-007 | 加入购物车的商品单选和全选无法正常切换 | 重要 | 2017-4-20  11:30:20 | 2017-4-21  12:30:59 | 代码逻辑问题。无法进行单选、全选切换 | 已解决 |
| YY-008 | 用户对加入购物车的商品进行修改时，总金额不同自动变化 | 重要 | 2017-4-21  16:20:55 | 2017-4-21  18:30:56 | 用户删除商品或者添加商品数量时，总金额应该实时变化 | 已解决 |
| YY-009 | 填写收货地址时，无法正常获取地址信息 | 重要 | 2017-4-22  10:30:15 | 2017-4-22  14:30:20 | 下载的全国json文件信息不全 | 已解决 |
| YY-010 | 登录注册时自动生成验证码 | 重要 | 2017-4-23  9:15:23 | 2017-4-23  12:23:10 | 自动生成的验证码不能验证 | 已解决 |

5.4 本章小结

测试环节是项目中必不可少的部分。本章主要罗列了一些前端开发中常见的Bug，了解这些常见的Bug可以明显的提高工作和测试效率。最后，我对前端测试方法进行了简单的论述，可以让我们了解PC端的本地测试方式和移动端的测试方式，并对常用的测试工具做了介绍。

结 论

近年来，随着互联网的高速发展，各大电商网站迅速发展。极大地方便了人么的生活。人们可以足不出户的完成对自己喜爱的商品的购买。因此对于人们第一眼看见的电商门户网站就有了更多的需求，网站是用户第一眼所见的东西。因此对于网站提供的功能，视觉感受都有了更高的要求。而对于前端开发者来说，对于我们的要求也越来越多。因为我们是和用户最近的开发者。

PC端的前端开发技术及理念已经普遍成熟，只要是应用JavaScript，CSS和HTML三种元素来构建桌面网页，考虑的最多问题便是网页在各种浏览器之间的兼容性，性能的考虑虽然也有很大的比重，但是随着硬件设施的更新，在网速越来越快的今天，网页性能问题也随之下降。但是对于我们前端开发者来说，不能只是通过代码完成页面的搭建。我们现在应该考虑更多的用户体验问题，在开发过程中，要将自己带入用户的角色。站在用户的角度上考虑问题，如果使自己搭建的桌面网页能是用户拥有最高的用户体验，这是我们首先应该思考的问题。

在互联网高速发展的大浪潮下，可以发现前端开发这个职业会越来越有前途，前端开发在企业中所占的比例也会越来越大。而互联网是一个技术更新迭代非常迅速的领域，并且前端是与最新技术最紧密的部分。Web前端开发的主要职务就是把网站的页面更好地呈现给用户，它是一份特殊的工作。看似简单，但其实它所涵盖的知识面非常广。作为一个前端开发者，不仅要掌握具体的技术，还要包含抽象的理念。毕竟我们是所有从事IT行业里面离用户最近的一环。用户是否用户好的用户体验，取决于我们的工作。

老话说的好，实践出真知，自己完成一个真实的项目，结合前辈们的研究，我们在设计和开发过程中遵循一定的流程，把前端的设计和实现工作进一步专业化、流程化和规范化，形成一套可行的规范，提高前端开发的工作效率。同时我们研究了产品的一些布局和交互设计，并对怎么实现这些设计和布局进行了探讨。我们还研究了怎么提高CSS的性能和网站的性能，提出了一些可行的建议，并且运用于项目中，最后开发出一个美观大方，简介有效，用户体验好的互联网网站。同时探讨了常见的前端开发兼容性Bug，并介绍了最常见的前端测试方法和测试工具。

致 谢

我的毕业设计已经完成了，在此我首先要感谢我的指导老师，曹洪宇老师，在进行毕业设计这漫长的半年中，曹老师对我进行了细致的指导。从一开始的分配任务，到后来的程序设计和完成论文，曹老师一直督促着我，指出我的不足，并且教会了我很多不懂的地方。

其次我要感谢以往每一位教过我的老师，系统开发是一项结合很多方面的大工程，它不是掌握一方面知识之后就能完成的，是以往教我的每一位老师的悉心教导，才让我如今作出了这样的系统。

最后我要感谢我的同学们，在这次的毕设中我们小组的同学帮助了我很多，比如曹老师要求我们互相检查对方的论文有哪里出错，我自己觉得毫无问题的论文，在同学的细心检查下，发现了不少的错误。同学们为我指出错误，然后我加以改正。

是各位老师的指导，以及我自身的努力，才能完成今天的毕设，感谢各位老师同学，让我的大学生活画上一个圆满的句号。

参考文献

[1] (美)David.Flanagan. JavaScript权威指南(第六版)[J]. 机械工业出版社 2012.3.

[2] JonathanChaffer/KarlSwedberg. 基础教程[J]. 人民邮电出版社 2009.11.

[3] (荷兰)Peter-Paul Koch/奇舞团. 移动Web手册[M]. 电子工业出版社 2015.5.

[4] 张磊鹏. 企业接口服务管控系统的设计与实现[D].《中国优秀硕士学位论文数据库》2013, (S2).

[5] Andy Budd, (英)Simon Collison, Cameron Mol, 陈剑瓯. 精通CSS：高级Web标准解决方案(第 2版)[J]. 人民邮电出版社 2010.5.

[6] 单东林，张晓菲. 锋利的jQuery[M]. 魏然人民邮电出版社 2009-6.

[7] 魏娜. Web 前端开发技术研究[A]. 苏州经贸职业技术学院, 苏州215009.

[8] 田岭. 深入理解CSS选择符的匹配方式[J]. 软件导刊 2011.

[9] 王政. Web 前端开发技术以及优化研究[A]. 扬州工业职业技术学院，江苏扬州225007.

[10] 朱聚豹. 移动Web前端性能与图片优化的研究[D]. 湖北：华中师范大学计算机学院 2013年 5月.

[11] Nilesh Jain, Lecturer. Review Of Different Responsive CSS Front-End Frameworks[A]. Mandsaur Institute of Technology,Mandsaur, Volume 5, No. 11, November 2014.

[12] Rodrigo Laiola Guimarães, Dick C. A. Bulterman, Pablo Cesar and Jack Jansen. Synchronizing Web Documents with Style[A].

[13] Jongmyung Choi, Youngho Lee and Kiyoung Kim. An HTML5-based Interactive E-book Reader[A]. Vol.8, No.2 2014.

附 录