

**学期项目二-Legou商城**

**姓 名：**邓敏

**班 级：**软件20336班

**任课教师：**杜灵、高晓琴老师

四川工商职业技术学院 制

目录

[项目开发过程理解 3](#_Toc30280)

[知识点总结 4](#_Toc3349)

[面试18个问题 7](#_Toc23122)

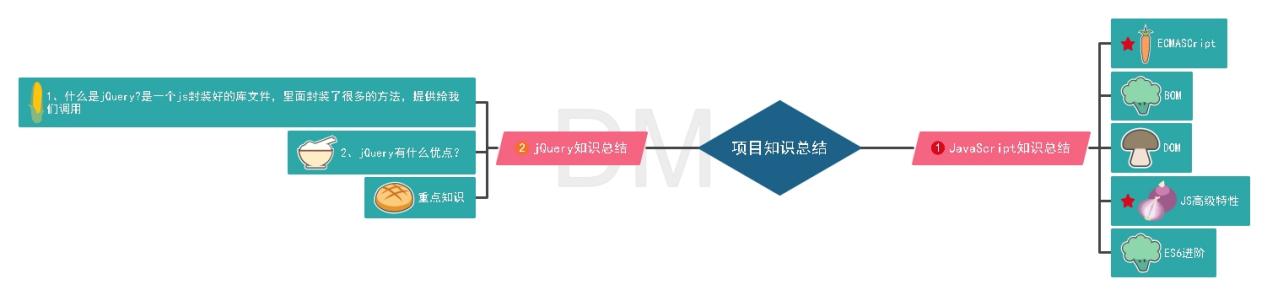
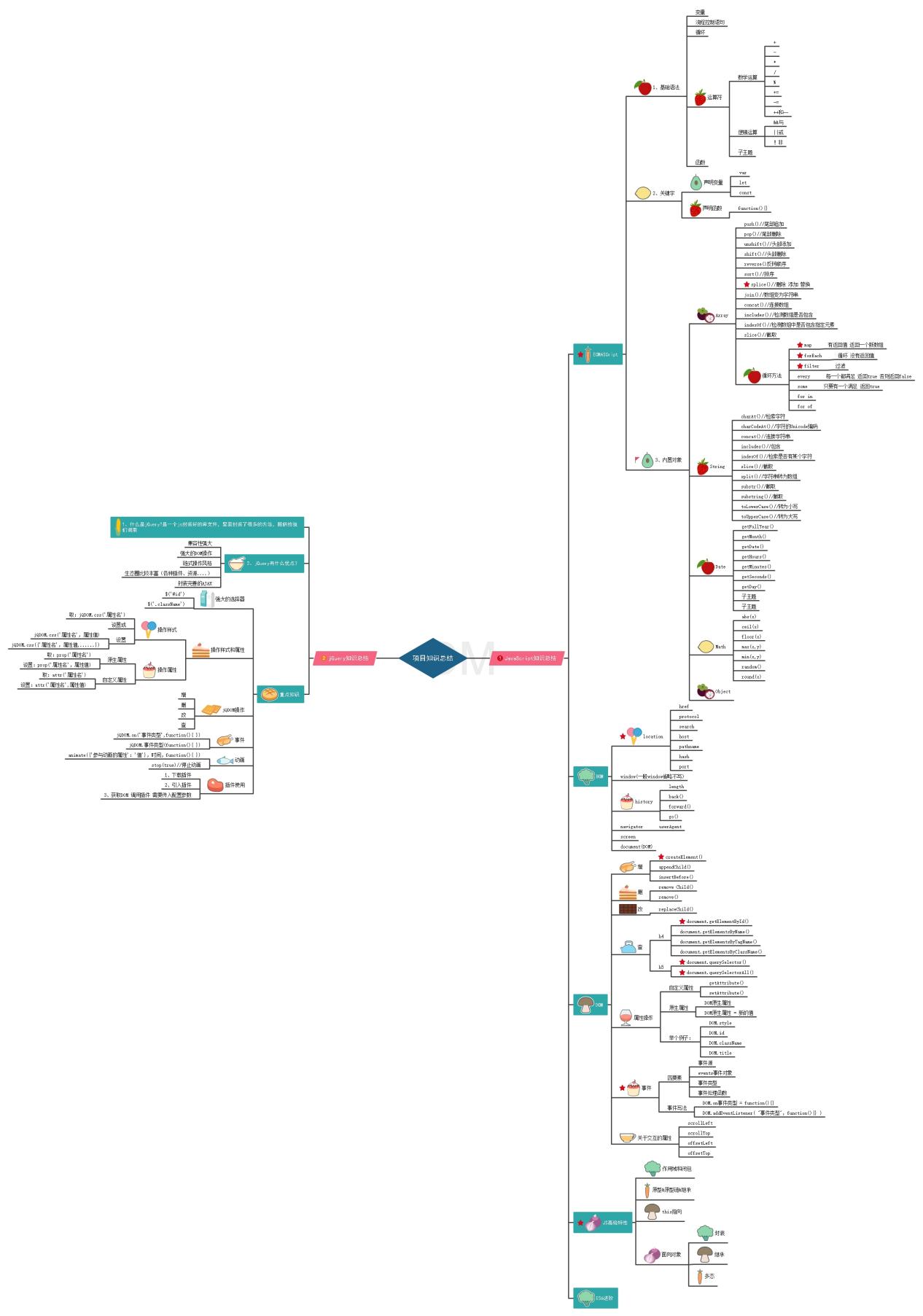
[商城项目开发遇到的问题及其解决方案 14](#_Toc7840)

# 项目开发过程理解

1. **项目介绍：**一个商城类型的项目，主要是一个平台。用户分为商家、卖家。商家可以在这个平台发布产品，出售产品，买家可以搜索，买商品，支付，评论。项目包含了：登录注册模块，首页展示模块，商品展示模块，购物车模块，产品详情模块，会员中心模块，……
2. **项目周期：**3~6个月
3. **人员配置：**项目经理：1  
   产品经理  
   ui设计：1  
   前端：2  
   后端：4  
   测试：1人（有些公司没有 有些公司测试是独立的）
4. **项目完整流程：**需求收集=>需求评审=>项目立项=>项目排期（甘特图）=>产品原型图  
   =>ui（前端后端）=>项目测试=>项目部署上线=>后期维护
5. **项目技术栈：**html/h5 css/css3 javaScript jQuery

**项目职责：（不低于4条）**  
1.负责了项目工程目录搭建  
2.高度还原的ui设计图  
3.首页模块以及所有前端交互  
4.购物车模块以及前端交互  
5.产品详情模块以及前端交互

# 知识点总结

1. **大致结构**
2. **详细总结**  
   

# 面试18个问题

1. **JS数据类型有哪些？（基本类型和引用类型）**a基本类型：undefined null boolean string number  
   b引用类型：object (object array function)
2. **如何转化类型？**parseFloat() //转为浮点数  
   parseInt() //转为整数  
   + 字符串 //变为字符串  
   - 数字 //变为数字

**3、数组从小到大排序？**<script type="text/javascript">

var Arr=[13,7,10,76,97,100,35];

var min;

for(var i=0; i<Arr.length; i++){

for(var j=i; j<Arr.length;j++){

if(Arr[i]>Arr[j]){

min=Arr[j];

Arr[j]=Arr[i];

Arr[i]=min;

}

}

}

document.write(Arr.toString()+"<br />");

</script>

**4、数组从大到小排序？**<script type="text/javascript">

//创建数组元素arr

var arr=[7,13,34,3,200,100,4];

//创建每次循环存储最大值得变量

var max;

//遍历数组，默认arr中的某一个元素为最大值，进行逐一比较

for(var i=0; i<arr.length; i++){

//外层循环一次，就拿arr[i] 和 内层循环arr.legend次的 arr[j] 做对比

for(var j=i; j<arr.length; j++){

if(arr[i]<arr[j]){

//如果arr[j]大就把此时的值赋值给最大值变量max

max=arr[j];

arr[j]=arr[i];

arr[i]=max;

}

}

}

document.write(arr.toString()+"<br>");

//200,100,34,13,7,4,3

</script>

1. **显示转换与隐式转换？**显示转换：人为的去改变数据的类型，又叫强制转换  
   隐式转换：在js中，当运算符在运算时，如果两边数据不统一，编译器会自动将运算符两边的数据做一个数据类型转换，转成一样的数据类型再计算
2. **Mouseover/mouseout与mouseenter/mouseleave的区别于联系？**a、mouseover/mouseout是标准事件，所有浏览器都支持；  
   mouseenter/mouseleave是IE5.5引入的特有事件，后来被DOM3标准采纳，现代浏览器也支持  
   b、mouseover/mouseout 是冒泡事件  
   mouseenter/mouseleave不冒泡  
   需要为多个元素监听鼠标移入/移出事件时，推荐使用mouseover/mouseout托管，提高性能

**7、forEach和map和filter的区别？**  
a、forEach()

用法：array.forEach(function(item，index){})

没有返回值，只是单纯的遍历

b、map()

用法：array.map(function(item，index){})

遍历并对其进行操作，返回新数组，原数组不变

c、filter()

用法：array.filter(function(item，index){})

筛选符合条件的元素并返回新数组，不会对空数组进行检测，原数组不变  
**区别：**

forEach和map都是遍历一个数组，但它们的返回值不同。forEach的返回值为undefined，不可以链式调用，而map回调函数的返回值会组成一个新数组，新数组的索引结构和原数组一致，原数组不变。

filter会返回原数组的一个子集，回调函数用于逻辑判断，返回 true则将当前元素添加到返回数组中，否则排除当前元素，原数组不变。

map和filter方法不会改变原数组。

这三个方法都不会对空数组进行检测或执行回调函数。

没有办法终止或者跳出forEach()循环，除非抛出异常，所以想执行一个数组是否满足什么条件，返回布尔值，可以用一般的for循环实现，或者用Array.every()或者Array.some()

1. **offsetWidth/offsetHeight、clientWidth/clientHeight与scrollWidth/scrollHeight的区别？**a、offsetWidth/offsetHeight返回值包含content + padding + border ，如果有滚动条，也不包含滚动条  
   b、clientWidth/clientHeight返回值只包含content + padding，如果有滚动条，也不包含滚动条  
   c、scrollWidth/scrollHeight返回值包含content + padding + 溢出内容的尺寸

**9、==与===的区别？**a、== 表示相等 （值相等）  
b、===表示恒等（类型和值都要相等）  
c、js在比较的时候如果是 == 会先做类型转换，再判断值得大小，如果是===类型和值必须都相等  
  
**10、Null和undefined的区别？** null用来表示尚未存在的对象，常用来表示函数企图返回一个不存在对象  
 undefined主要指定义了变量，但是并未赋值  
 Null(not a Number)不是一个明确数值的数字类型  
 ECMAScript 认为undefined是从null派生出来的，他们的值是一样的，但是类型却不一样。所以  
 null == undefined //true  
 null === undefined //false

**11、字符串操作方法（8个）**  1、indexOf()方法返回字符串中指定文本首次出现的索引（位置）  
 2、slice()方法提取字符串的某个部分并在新字符串中返回被提取的部分

3、replace()方法用另一个值替换在字符串中指定的值

4、toUpperCase()把字符串转换为大写

5、concat()连接两个或多个字符串

6、trim()方法删除字符串两端的空白符

7、charAt()方法返回字符串中指定下标（位置）的字符串

8、charCodeAt()方法返回字符串中指定索引的字符unicode编码

9、split()将字符串转换为数组

**12、数组操作方法（10个）**1、concat() 方法用于连接两个或多个数组。该方法不会改变现有的数组，仅会返回被连接数组的一个副本。  
2、join() 方法用于把数组中的所有元素放入一个字符串。元素是通过指定的分隔符进行分隔的，默认使用','号分割，不改变原数组。  
3、push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。末尾添加，返回的是长度，会改变原数组。  
4、pop() 方法用于删除并返回数组的最后一个元素。返回最后一个元素，会改变原数组。

5、shift() 方法用于把数组的第一个元素从其中删除，并返回第一个元素的值。返回第一个元素，改变原数组。  
 6、unshift() 方法可向数组的开头添加一个或更多元素，并返回新的长度。返回新长度，改变原数组  
 7、返回一个新的数组，包含从 start 到 end （不包括该元素）的 arrayObject 中的元素。返回选定的元素，该方法不会修改原数组。  
 8、splice() 方法可删除从 index 处开始的零个或多个元素，并且用参数列表中声明的一个或多个值来替换那些被删除的元素。如果从 arrayObject 中删除了元素，则返回的是含有被删除的元素的数组。splice() 方法会直接对数组进行修改。  
 9、reverse() 方法用于颠倒数组中元素的顺序。返回的是颠倒后的数组，会改变原数组。  
 10、fill() 方法用新元素替换掉数组内的元素，可以指定替换下标范围。

**13、JS中定时器有哪些？他们的区别及用法是什么？**1、setTimeout() 方法用于在指定的毫秒数后调用函数或计算表达式。

2、setInterval() 方法可按照指定的周期（以毫秒计）来调用函数或计算表达式。  
setInterval() 方法会不停地调用函数，直到 clearInterval() 被调用或窗口被关闭。由 setInterval() 返回的 ID 值可用作 clearInterval() 方法的参数。  
 3、setImmediate，任务队列执行完成后马上运行指定代码

如下代码演示：  
var timer = setInterval(function() {  
console.log('test');  
}, 1000);  
setTimeout(function() {  
clearInterval(timer);  
}, 3000);

**14、document.write和innerHTML的区别？** document.write是直接写入到页面的内容流，如果在写之前没有调用document.open, 浏览器会自动调用open。每次写完关闭之后重新调用该函数，会导致页面被重写。  
 innerHTML则是DOM页面元素的一个属性，代表该元素的html内容。你可以精确到某一个具体的元素来进行更改。如果想修改document的内容，则需要修改document.documentElement.innerElement。  
 innerHTML很多情况下都优于document.write，其原因在于其允许更精确的控制要刷新页面的那一个部分。

**15、解释一下DOM和BOM？** 1、BOM是Browser Object Model的缩写，即浏览器对象模型。

BOM和浏览器关系密切。浏览器的很多东西可以通过JavaScript控制的，例如打开新窗口、打开新选项卡（标签页）、关闭页面，把网页设为主页，或加入收藏夹，等等…这些涉及到的对象就是BOM。

2、DOM是Document Object Model的缩写，即文档对象模型。

DOM和文档有关，这里的文档指的是网页，也就是HTML文档。网页是由服务器发送给客户端浏览器的，无论用什么浏览器，接收到的HTML都是一样的，所以DOM和浏览器无关，它关注的是网页本身的内容。由于和浏览器关系不大，所以标准就好定了。

**16、jQuery一个对象可以同时绑定多个事件，这是如何实现的？** bind on delegate live 进行多事件绑定的原理  
 jQuery每调用一个方法，或绑定事件，都返回this(当前的dom)，所以可以链式操作。绑定多个事件

**17、针对jQuery的优化方法？** 缓存频繁操作DOM对象  
 尽量使用ID选择器代替类选择器  
 总是从#ID选择器来继承  
 尽量使用链式操作  
 在绑定事件上使用时间委托  
 采用的jQuery的内部函数数据（）来存储数据  
 使用最新版本的jQuery

**18、jQuery常用选择器？**

**1.基本选择器** #id $（"#test").css("background","#bbffaa")//选取id为test的元素

element $("p") ;//选取所有的<p>元素

\* $("\*");//选取所有元素

.class $(".test");//选取类名为test的元素

selector1,selector2 $("div,p.myClass,span");//组合选择  
  
**2.层次选择器**$("div span")//选取<div>里所有<span>元素（所有后代元素）

$("div>span")//选取<div>元素下元素名为<span>的子元素（只限于子元素）

$(".one+div")//选取class 为one的下一个<div>同辈元素

等价于$(".one").next("div")

$("#two~div")//选取id为two的元素后面的所有<div>同辈元素

等价于$("#two").nextAll("div")

**3.过滤选择器（以“：”开头）**  
**a、基本过滤**:first 选取第一个元素: $("div:first") 选区所有<div>元素中第一个<div>元素

:last 选取最后一个元素 $("div:last")

:not(selector) 去除所有给定选择器匹配的元素 $("input:not(.myClass)") 选取class不是myClass的<input>元素

:even 选取索引数是偶数的元素 索引从0开始 $("div:even")

:odd 选取索引数是奇数的元素 索引从0开始 $("div:odd")

:eq(index)选取索引数等于index的元素 $("div:eq(2)")

:gt(index)选取索引数大于index的元素 $("div:gt(2)")

:lt(index)选取索引数小于index的元素 $("div:lt(2)")

:header 选取所有标题元素<h1>..<h6> $(":header")

:animated 选取所有当前正在执行动画的的所有元素 $("div:animated")选取正在执行动画的<div>元素

:focus 选取当前获得焦点的元素 $(":focus")  
  
**b、内容过滤**:contains(text)选取含有文本内容为text的元素 $("div:contains('我')")选取含有文本"我"的<div>元素

:empty 选取不包含子元素或者文本的空元素 $("div:empty")选取不包含子元素（包括文本元素）的<div>空元素

:has(selector)选取含有选择器所匹配的元素的元素 $("div:has(p)")选取含有<p>元素的<div>元素

:parent 选取还有子元素或文本的元素 $("div:parent")选取拥有子元素（包括文本元素）的<div>元素  
  
**c、可见性过滤**:hidden 选取所有不可见元素 $(:hidden)选取所有不可见元素 包括<input type="hidden"> <div style="display:none"> 和<div style="visibility:hidden">等 如果只想选取<input>元素，可以使用 $("input:hidden")

:visible 选取所有可见元素 $(div:visible) 选取所有可见的<div>元素

例如 显示隐藏的<div>元素 $(div:hidden).show(3000)  
  
**d、属性过滤**[attribute] 选取拥有此属性的元素 $("div[id]")选取拥有属性id的元素

[attribute=value] 选取属性值为value的元素 $("div[title=test]") 选取属性title为“test”的<div>元素

[attribute!=value] 选取属性的值不等于value的元素 $("div[title!=test]")

[attribute^=value] 选取属性的值以value开始的元素 $("div[title^=te]")选取属性title的值以te开始的div元素

[attribute$=value] 选取属性的值以value结束的元素 $("div[title$=est]")选取属性title的值以est结束的div元素

[attribute\*=value] 选取属性的值含有value的元素 $("div[title\*=es]")选取属性title的值含有es的div元素

[attribute|=value] 选取属性等于给定字符串或以该字符串为前缀（该字符串后跟一个连字符"-"）的元素 $('div[title|="en"]')

[attribute~=value] 选取属性用空格分隔的值中包含一个给定值的元素 $('div[title~="uk"]')

[attribute1][attribute2][attributeN] 用属性选择器合并成一个复合属性选择器，满足多个条件。每选择一次，缩小一次范围

$("div[id][title$='test']")  
  
**e、子元素过滤**:nth-child(index/evenn/odd/equation) 选取每个父元素下得第index个子元素或者奇偶元素（index从1算起）

解析：:eq(index)只匹配一个元素，而:nth-child将为每一个父元素匹配子元素，并且:nth-child(index)的index是从1开始的，而:eq(index)是从0开始的

:first-child 选取每个父元素的第一个子元素 集合元素

解析：:first只返回单个元素，而:first-child选择符将为每个父元素匹配第一个子元素。例如$("ul li:first-child");选取每个<ul>中第一个<li>元素

:last-child 选取每个父元素的最后一个子元素

解析：:last只返回单个元素，:last-child选择符将为每个父元素匹配最后一个子元素 例如$("ul li:last-child")；选择<ul>中最后一个<li>元素

:only-child 如果某个元素是他父元素中唯一的子元素，那么将会被匹配。如果父元素中含有其他元素，则不会被匹配 $("ul li:only-child")在<ul>中选取是唯一子元素的<li>元素  
  
**f、表单对象属性过滤**  
:enabled 选取所有可用元素 $("#form1:enabled") 选取id为form1的表单内所有可用元素

:disabled 选取所有不可用元素 $("#form1:disabled")

:checked 选取所有被选中的元素（单选框，复选框）$("input:checked")选取所有被选中的<input>元素

:selected 选取所有被选中的选项元素（下拉列表）

$("select option:selected") 选取所有被选中的选项元素  
  
g、**表单选择器**:input  
:text  
:password  
:radio  
:checkbox  
:submit  
:image  
:reset  
:button  
:file :hidden

# 商城项目开发遇到的问题及其解决方案

a问题：

1. html
2. css
3. js

b解决方案：

html：

1、审查元素

2、注释（删除）排错

3、布局大部分有问题都是浮动和定位的问题

css:

1、审查元素，微调，注意看样式

2、发现有样式，但是不生效（选错了 优先级）

3、样式选择的范围过大（用子选择器）

ul.myul

li

ul li

.myul > li

4、样式能写类，尽量全部写类，少写标签选择器

js:

1、打印法（理论上，JS每一步都可以打印，如果打印·的不是你预想的结果，就有问题）

2、数据代入法，把数据代入你写的函数代码，运行，看是否是预想结果。

3、把错误翻译成中文，基本看懂意思，就能解决问题

4、打断点

调到下一个断点，单步执行，遇到函数，用结果，不进入函数

进函数

出函数

单步