**学期项目二-Legou 商城**

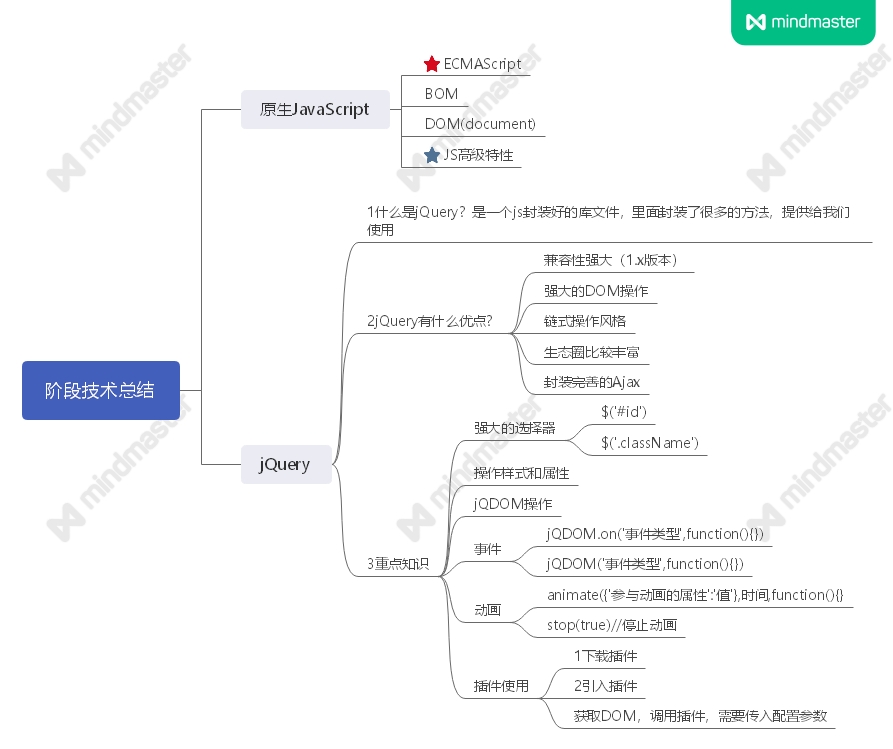
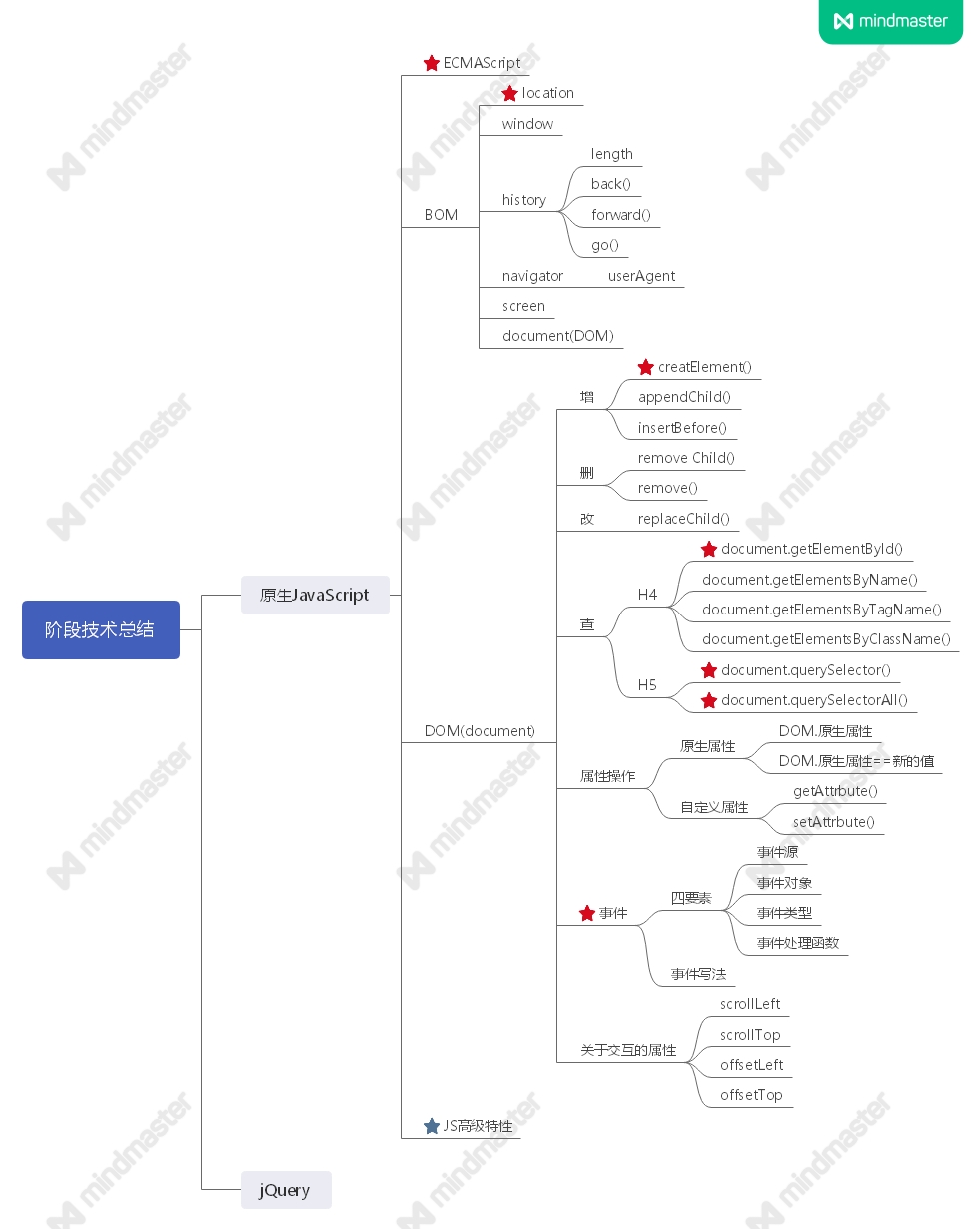
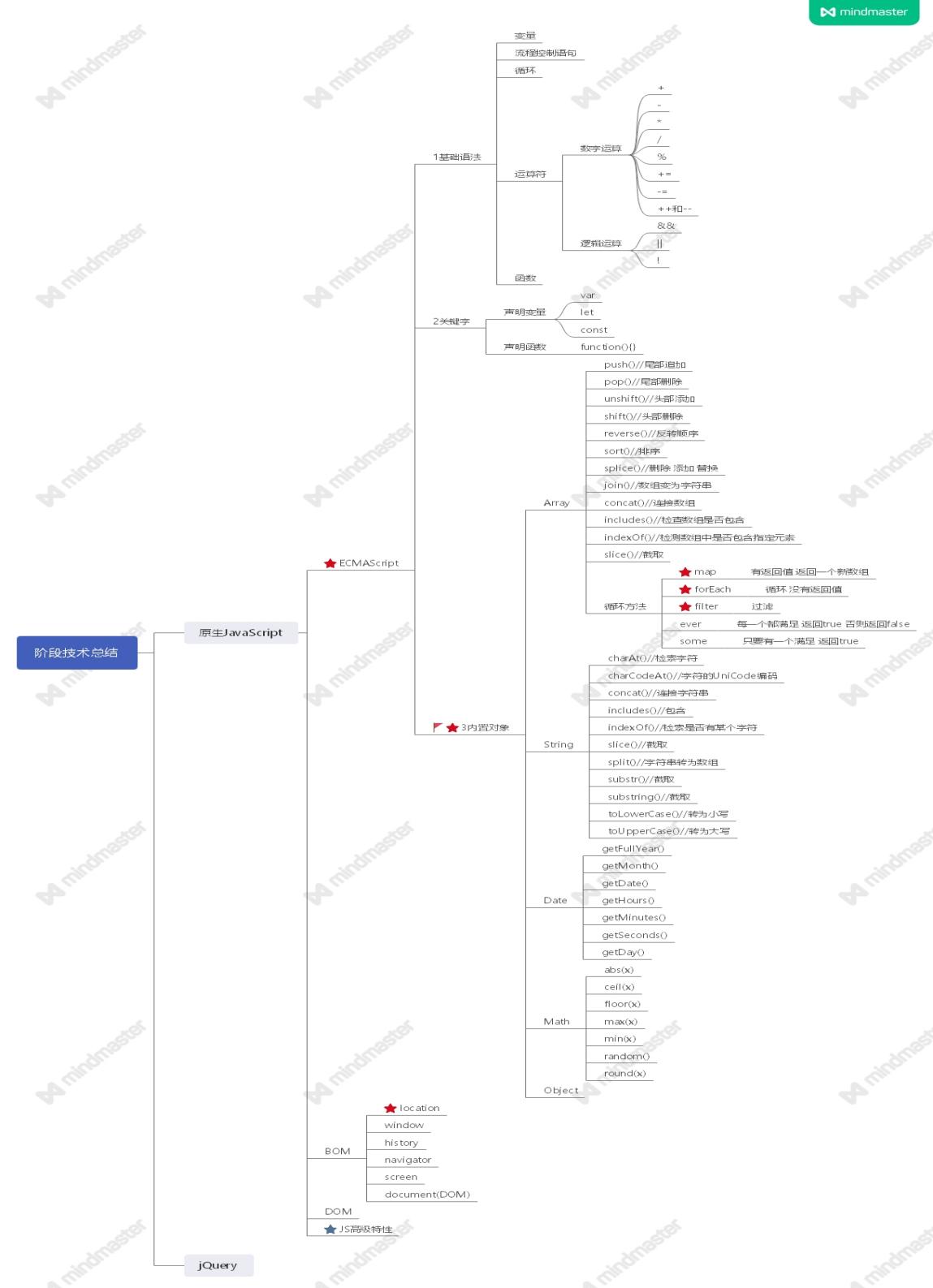
**----------------孔雅诗**

1. **项目开发过程理解...........................................................2**
2. **知识点总结......................................................................3**
3. **面试18个问题................................................................6**
4. **商城项目开发遇到的问题及其解决方案........................16**

**项目开发过程理解**

**完成项目开发首先要注意的是明确任务需求，对需求有大概的了解，然后根据需求先对页面进行基础排版，再对需要代码支持的板块进行了解，大概需要什么代码来达成这种效果，然后再进行网页的制作**

**知识点总结**



1、JS数据类型有哪些？

JavaScript的数据类型共有六种。. 它们分别是**undefined、null、boolean、number、string、object**. 它们共分为两大类，分别为：. 基本类型：字符串（String）、数字 (Number)、布尔 (Boolean)、对空（Null）、未定义（Undefined）. 引用类型：对象 (Object)、数组 (Array)、函数 (Function)

1. 如何转化类型？

在**JS** 中只有 3 **种类型的转换**: **转化**为 Number **类型**：Number () / parseFloat () / parseInt () **转化**为 String 类型：String () / toString ()

1. 数组从小到大排序？

arr1.sort((a,b)=>{

// 每相邻的两个值如果值最小或者最大则将其返回

// 第一个参数为每一次循环的值，但是不包括第一个

// 第二个参数为每一次循环的值，但是不包括最后一个

// 从第二个值开始减自己的前一个值，将这个结果进行返回并且按升序的顺序排序

return a-b;

// 从第一个值开始减自己的后一个值，将这个结果进行返回并且按降序的顺序排序

// return b-a;

})

console.log(arr1) //[5, 7, 8, 9, 11, 15]

1. 数字从大到小排序？

原理同上，代码改为arr.sort(function(a,b){returnb-a}

1. 显式转换与隐式转换

**js的显式转换和隐式转换都是数据类型的转换**；

显式转换就是通过js定义的方法来转换数据类型；  
主要方法有: Number(), String(), parseInt(), parseFloat(), Boolean(),toString();

js的数据类型是弱类型的，即可以给变量赋值为任意的数据类型，当进行运算时可能会遇到数字与对象的相加，这是就会触发隐式转换。

1. mouseove/mouseoutr和mouseenter/mouseleave的区别

**mouseover：当鼠标移入元素或其子元素都会触发事件，所以有一个重复触发，冒泡过程。 对应的移除事件是mouseout mouseenter:当鼠标移除元素本身（不包含元素的子元素）会触发事件，也就是不会冒泡，对应的移除事件是mouseleave** 1. 是否支持冒泡 先看一张图，对这两件事有一个简单直观的感受。 当鼠标从元素的边界之外移入元素的边界之内时，事件被触发。 而鼠标本身在元素边界内时，要触发该事件，必须先将鼠标移出元素边界外，再次移入才能触发。 由于mouseenter不支持事件冒泡，导致在一个元素的子元素上进入或离开的时候会触发其mouseover和mouseout事件，但不会触发mouseenter和mouseleave事件。

1. forEach和map和filter的区别

### forEach遍历数组，参数为一个回调函数，回调函数接收三个参数，当前元素，元素索引，整个数组， forEach 在对 item 进行修改的时候，如果 item 是原始类型的值，item 对应的 的内存地址实际并没有变化，

如果 item 是 引用类型的值，item 对应多的内存地址也没有变化，但是对应的值，已经重写了  
map与forEach类似，遍历数组，但其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。

1. offsetWidth/offsetHeight、clienWidth/clirnHeight与scrollWidth/scrollHeight的区别

offsetWidth/offsetHeight返回值包含content + padding + border ，如果有滚动条，也不包含滚动条

clientWidth/clientHeight返回值只包含content + padding，如果有滚动条，也不包含滚动条

scrollWidth/scrollHeight返回值包含content + padding + 溢出内容的尺寸

1. ==与===的区别
2. 对于 string、number 等基础类型，== 和 === 是有区别的 a）**不同类型间比较，== 之比较 "转化成同一类型后的值" 看 "值" 是否相等，=== 如果类型不同，其结果就是不等**。

2、对于 Array,Object 等高级类型，== 和 === 是没有区别的

进行 "指针地址" 比较

3、基础类型与高级类型，== 和 === 是有区别的

* a）对于 ==，将高级转化为基础类型，进行 "值" 比较
* b）因为类型不同，=== 结果为 false

1. != 为 == 的非运算，!== 为 === 的非运算
2. null和undefined的区别

null和undefined之间的主要区别在于它们**被转换为原始类型的方式**。 在null上执行算术转换时，确定的值为0可以使用以下代码片段验证此转换。

1. 字符串操作方法

concat() – 将两个或多个字符的文本组合起来，返回一个新的字符串。

indexOf() – 返回字符串中一个子串第一处出现的索引。如果没有匹配项，返回 -1 。

charAt() – 返回指定位置的字符。

lastIndexOf() – 返回字符串中一个子串最后一处出现的索引，如果没有匹配项，返回 -1 。

match() – 检查一个字符串是否匹配一个正则表达式。

substring() – 返回字符串的一个子串。传入参数是起始位置和结束位置。

replace() – 用来查找匹配一个正则表达式的字符串，然后使用新字符串代替匹配的字符串。

search() – 执行一个正则表达式匹配查找。如果查找成功，返回字符串中匹配的索引值。否则返回 -1 。

slice() – 提取字符串的一部分，并返回一个新字符串。

split() – 通过将字符串划分成子串，将一个字符串做成一个字符串数组。

length – 返回字符串的长度，所谓字符串的长度是指其包含的字符的个数。

toLowerCase() – 将整个字符串转成小写字母。

toUpperCase() – 将整个字符串转成大写字母。

1. 数组操作方法

##### 1. join()

功能：将数组中所有元素都转化为字符串并连接在一起。

##### 2. reverse()

功能：将数组中的元素颠倒顺序。

##### 3. concat()

功能：数组拼接的功能 ,返回新数组，原数组不受影响。

##### 4. slice()

截取数组生成新数组，原数组不受影响。  
返回的数组包含第一个参数指定的位置和所有到但不含第二个参数指定位置之间的所有元素。如果为负数，表示相对于数组中最后一个元素的位置。如果只有一个参数，表示到数组末尾。

##### 5. splice()

功能：从数组中删除元素、插入元素到数组中或者同时完成这两种操作。  
输入：第一个参数为指定插入或删除的起始位置，第二个参数为要删除的个数。之后的参数表示需要插入到数组中的元素 。如果只有一个参数，默认删除参数后边的所有元素。  
输出：返回一个由删除元素组成的数组。  
注意：新建了一个数组，并修改了原数组

##### 6. push()

在数组末尾添加一个或多个元素，并返回新数组长度

##### 7. pop()

从数组末尾删除1个元素(删且只删除1个), 并返回 被删除的元素

##### 8. shift()

在数组开始添加一个或多个元素，并返回新数组长度

##### 9. unshift()

在数组开始删除一个元素(删且只删除1个),并返回 被删除的元素

##### 10. toString()和toLocaleString()

将数组的每个元素转化为字符串，并且输入用逗号分隔的字符串列表。功能类似join();

##### 11. indexOf()和lastIndexOf()

indexOf() 两个参数：要查找的项和（可选的）表示查找起点位置的索引。其中， 从数组的开头（位置 0）开始向后查找。没找到返回-1. 返回查找项的索引值  
lastIndexOf() 从数组的末尾开始向前查找。返回查找项的索引值(索引值永远是正序的索引值),没找到返回-1

1. js中定时器有哪些？它们的区别以及用法是什么？

**javascript**里的**定时器有两种**：1、**单**次**定时器**，使用setTimeout ()方法定义，可以在规定时间（以毫秒计）过后执行一次代码块；2、循环**定时器**，使用setInterval ()方法定义，按照指定的周期（以毫秒计）来重复执行某些代码。

1. document.write和innerHTML的区别？

1.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](https://www.fly63.com/tag/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "https://www.fly63.com/article/detial/_blank)会自动调用open。每次写完关闭之后重新调用该[函数](https://www.fly63.com/tag/%E5%87%BD%E6%95%B0" \t "https://www.fly63.com/article/detial/_blank)，会导致页面被重写。

inner1.write是[dom](https://www.fly63.com/tag/dom" \t "https://www.fly63.com/article/detial/_blank)方法,向文档写入HTML表达式或JavaScript[代码](https://www.fly63.com/tag/%E4%BB%A3%E7%A0%81" \t "https://www.fly63.com/article/detial/_blank)，可列出多个参数，参数被顺序添加到文档中 ；innerHTML是DOM属性,设置或返回调用元素开始结束标签之间的HTML元素。

2.两者都可向页面输出内容,innerHTML比document.write更灵活。  
当文档加载时调用document.write直接向页面输出内容，文档加载结束后调用document.write输出内容会重写整个页面。通常按照两种的方式使用 write() 方法：一是在使用该方在文档中输出 HTML，二是在调用该方法的的窗口之外的窗口、[框架](https://www.fly63.com/tag/%E6%A1%86%E6%9E%B6" \t "https://www.fly63.com/article/detial/_blank)中产生新文档（务必使用close关闭文档）。  
在读模式下，innerHTML属性返回与调用元素的所有子节点对应的HTML标记，在写模式下，innerHTML会根据指定的值创建新的DOM树替换调用元素原先的所有子节点。  
3.两者都可动态包含外部资源如JavaScript文件  
通过document.write插入<script>元素会自动执行其中的脚本；  
大多数浏览器中，通过innerHTML插入<script>元素并不会执行其中的脚本。

1. 解释一下DOM和BOM?

一个**DOM 会有一个根对象，这个对象通常就是 document**。 而 BOM 除了可以访问文档中的组件之外，还可以访问浏览器的组件，比如问题描述中的 navigator（导航条）、history（历史记录）等等。 在这种 「XOM」的模型中，最应该理解的就是 Object Model。

1. jQuery可以一个对象绑定多个事件，这是如何实现的？

jQuery 可以给一个对象同时绑定多个事件，低层实现方式是使用 addEventListner 或 attachEvent 兼容不同的浏览器实现事件的绑定，这样可以给同一个对象注册多个事件。

17、针对jQuery的优化方法？

1.使用选择器

尽量选用id选择器和元素选择器:因为遇到这些选择器的时候,jquery会自动调用浏览器的原生方法比如(document.getElementById()),所以他们的执行速度快.

2.尽量不要使用 ".live()" 方法

这是jquery1.3.1版本之后新增加的方法,这个方法的功能就是为新增DOM元素绑定事件.但对于效率来说,这个方法比较占用资源

3.使用data()

尽量使用data()方法储存变量,因为这是jquery的内部函数 data()来存储状态.

1. jQuery的常用选择器

**基本选择器**是JQuery最常用的选择器，也是最简单的选择器，它通过元素id、class和标签名来查找DOM元素。 注意：在网页中id只能使用一次，即id具有唯一性，但class是允许重复使用的。 层次选择器通过DOM元素间的层次关系来获取元素，主要的层次关系包括父子、后代、相邻、兄弟关系。 说明：只有这个方法返回的是JQuery对象才能进行链式操作。

|  |  |
| --- | --- |
| $("#myELement") | 选择id值等于myElement的元素，id值不能重复在文档中只能有一个id值是myElement所以得到的是唯一元素 |
| $("div") | 选择所有的div标签元素，返回div元素数组 |
| $(".myClass") | 选择使用myClass类的css的所有元素 |
| $("\*") | 选择文档中的所有的元素，可以运用多种的选择方式进行联合选择：例如$("#myELement,div,.myclass") |

**商城项目开发遇到的问题及其解决方案**

**主要遇到的问题是购物车代码问题，前面写的时候全选一直有问题，后面进行了重写还是有问题，后面发现是代码打错了，对原先声明的对象进行修改的时候有部分代码没有更改。还有就是电子书轮播图鼠标移入界面就会出错，看了几个同学的代码也没明白哪里有问题，至今没解决。**