**校叮当校园app项目代码清单**



|  |  |  |
| --- | --- | --- |
| 文件状态  【 】草稿  【 】正式发布  【√】正在修改 | 文件标识 | SE-2018春-G15-校叮当校园app需求分析报告 |
| 当前版本 | 0.1.0 |
| 作者 | 陈佳敏、徐毓茜、江亮儒、马益亮 |
| 完成日期 | 2018-5-28 |

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 陈佳敏、徐毓茜、江亮儒、马益亮 | 2018-5-26至2018-5-28 | 完成需求分析初稿，查找模板。 |

**目录**

[一、 概述 4](#_Toc12763)

[1.1需求概述 4](#_Toc26198)

[1.2参考资料 4](#_Toc14103)

[二、 详细设计描述 5](#_Toc26989)

[2.1功能结构（HIPO图、PAD图） 5](#_Toc22285)

[修改个人信息： 9](#_Toc26869)

[三、数据 10](#_Toc10451)

[3.1 E-R图 10](#_Toc13992)

[3.2数据表结构 11](#_Toc23118)

[User表 11](#_Toc10143)

[Commodity表 11](#_Toc30672)

[Order表 12](#_Toc31743)

[Message表 13](#_Toc31139)

[Collection表 13](#_Toc26832)

[四、命名规范 13](#_Toc3818)

[五、编码规则 14](#_Toc32232)

[5.1javascript书写位置 14](#_Toc6268)

[5.2javascript命名规范 14](#_Toc12237)

[5.3注释 15](#_Toc7075)

[5.4严格模式 16](#_Toc19957)

[5.5模式 16](#_Toc29681)

[5.6编写规范 17](#_Toc12249)

[5.7变量 18](#_Toc1150)

[三、 用户模块代码 19](#_Toc30632)

**[[1]](#footnote-0)**

# 概述

## 1.1需求概述

程序将分为以下四大模块：前端信息聚集和检索模块，用户模块，管理员模块，信息库模块。前端信息聚集和检索模块是程序的核心模块之一，它会呈现管理员所要给用户展现的内容以及检索用户需要的信息；用户模块和管理员模块对信息拥有不同的权限；信息库模块更新和存放程序将会用到的数据。

## 1.2参考资料

·《软件工程导论》 清华大学出版社 作者：张海藩等

·SE2017秋-G07-项目计划 作者：陈佳敏、徐毓茜、江亮儒、马益亮

·SE2017秋-G07-可行性计划 作者：陈佳敏、徐毓茜、江亮儒、马益亮

·SE2017秋-G07-需求分析报告 作者：陈佳敏、徐毓茜、江亮儒、马益亮

·SE2017秋-G07-总体设计 作者：陈佳敏、徐毓茜、江亮儒、马益亮

·SE2017秋-G07-详细设计 作者：陈佳敏、徐毓茜、江亮儒、马益亮

·http://blog.csdn.net/baidu\_32262373/article/details/52801121 js编码规范

# 详细设计描述

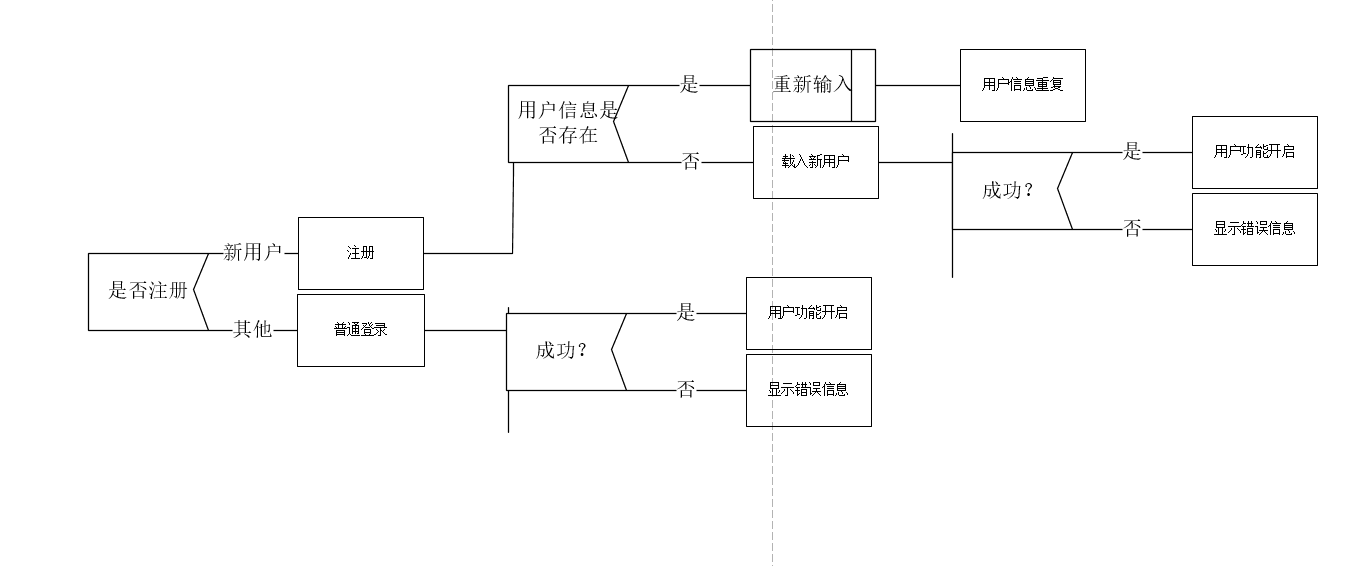
## 2.1功能结构（HIPO图、PAD图）

|  |  |
| --- | --- |
| 系统管理 | 管理员功能：对用户发出警告，从用户信息表中删除非法用户。这项功能只能为管理员所有。 |
| 用户管理 | 这个模块实现了用户请求交易，发布、更新交易信息的功能，并使用户能够更新自己的注册信息以及基本信息。 |
| 交易管理 | 在用户交易管理中，根据用户的提供的交易物品生成了“出售信息表”，用户可以随时对自己发布的信息进行更改或删除。在交易的过程中，系统会根据物品的出售信息目录供用户浏览参考。对于没有明显出售目标的用户，也可以浏览由出售信息表生成的出售物品目录 |
| 出售管理 | 用户发出出售请求后，系统会自动检查用户是否已登录。  用户录入出售信息（包括出售物品名称（gods），出售物品价格（sellprice），出售物品种类（category），出售物品描述（selldes），联系电话（contactp），联系QQ（contactq），联系微信（contactw））。其中物品名称、价格、种类为必填字段。联系方式为电话、微信或QQ，考虑到用户可能不愿意留下自己的电话、微信或QQ，我们将联系方式设为可选字段。当有交易请求时，系统通知用户。  经用户检查录入信息正确无误，生成出售物品信息单。  系统将出售信息纳入出售信息表，并根据出售信息表上的内容发布出售信息 |
| 买卖信息更新管理 | 用户可以随时更新自己的买卖信息。 |
| 搜索 | 用户录入搜索信息开始搜索，系统匹配后生成搜索结果。 |
| 用户信息管理 | 用户信息管理包括更改用户注册手机、更改用户密码、更改用户基本信息；交易管理包括出售管理这些管理操作，主要针对用户信息表进行修改，每个用户只有在登陆后才能修改自己的信息 |
| 更新注册手机 | 用户修改自己的注册手机信息时，必须输入原来的账户信息（注册手机和密码），经系统审核无误后，用户可以修改注册手机 |
| 更新用户密码 | 用户修改密码时，需要输入原密码，并两次输入新密码，经系统审核无误后，用户可以修改密码 |
| 更新用户基本信息 | 用户可以随时更新自己的买卖信息 |
| 用户注册 | 用户录入账户信息，进行有效性检查，生成用户信息表，返回登录界面进行登录 |

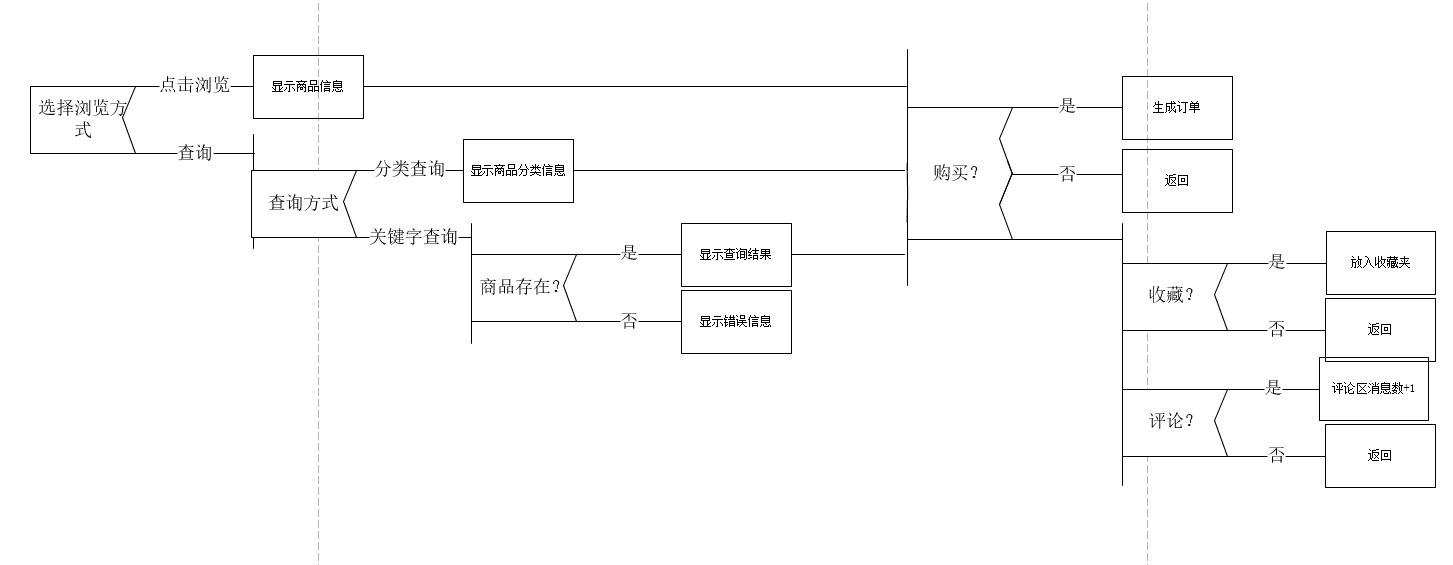
HIPO图

PAD图

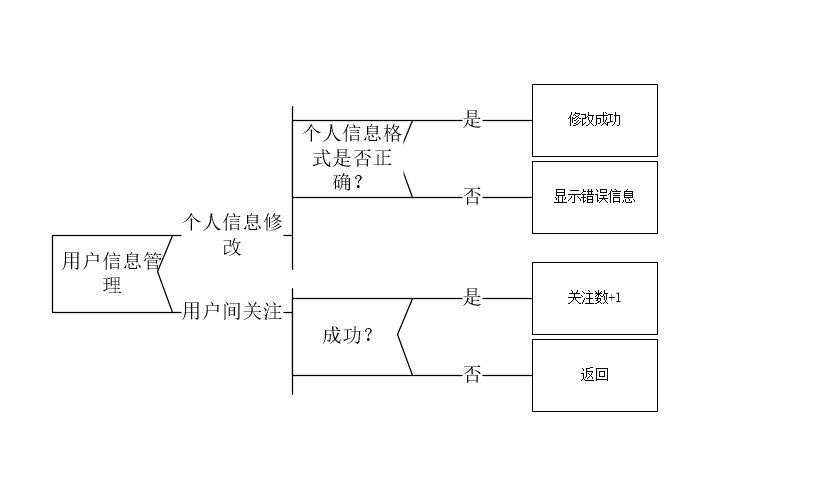
登录注册：



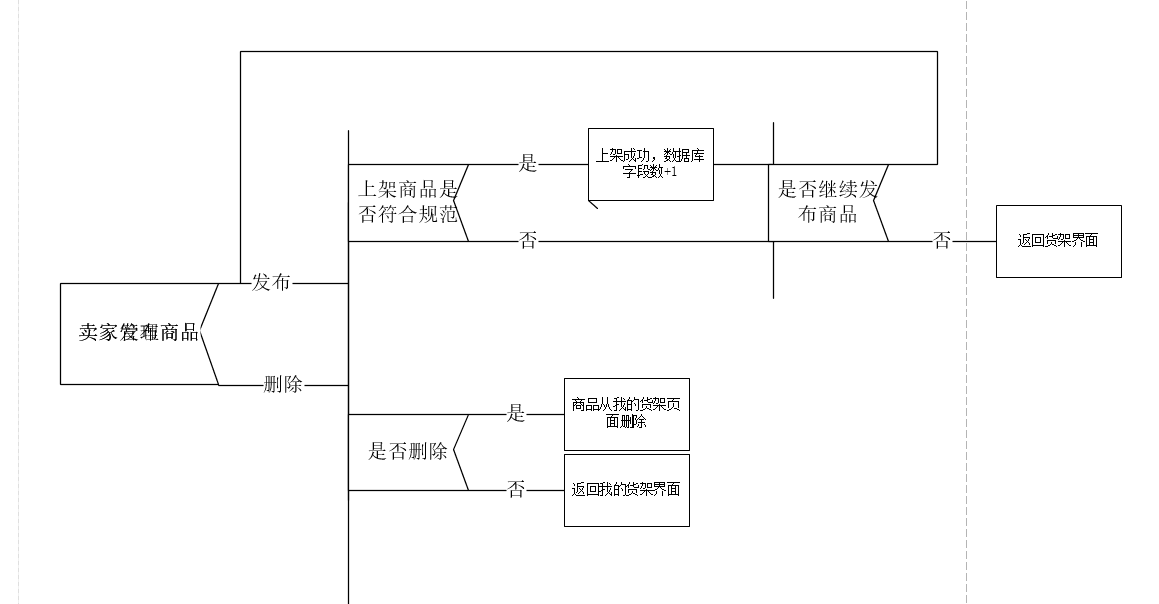
搜索、收藏、购买：



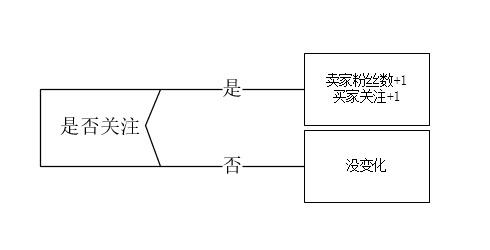
# 修改个人信息：



发布：

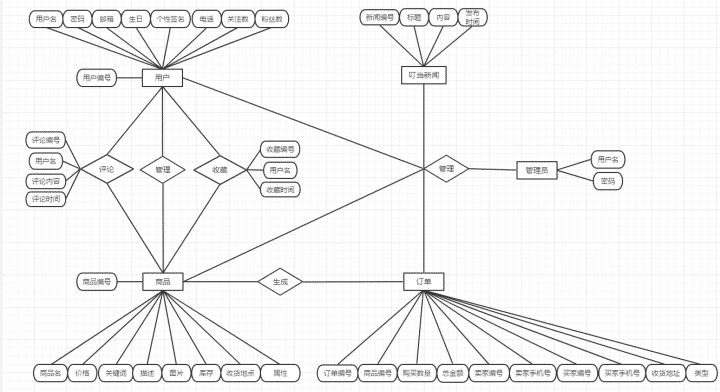


关注：



# 三、数据

## 3.1 E-R图



## 3.2数据表结构

### User表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User表 | | | | | |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 帐号 | user\_id | varchar | 不能 | 主键 |  |
| 密码 | user\_pw | varchar | 不能 |  |  |
| 姓名 | user\_name | varchar | 不能 |  |  |
| 性别 | user\_sex | varchar | 不能 |  | 可以设置为保密 |
| 生日 | user\_birth |  | 不能 |  |  |
| 手机号 | user\_phone | varchar | 不能 |  | 用于找回帐号，以及与交易对象进行联系 |
| 邮箱地址 | user\_email | varchar | 不能 |  | 用于找回帐号，以及与交易对象进行联系 |
| 拥有积分 | user\_score | int | 不能 |  | 上架商品需要积分 |
| 学号 | user\_sid | varchar |  |  | 未知 |
| 个人简述 | user\_resume | varchar |  |  | 简单的表述自己 |
| 照片 | user\_picture | varchar |  |  | 数据库只保存图片的地址，图片保存在硬盘里 |

### Commodity表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| commodity | | | | | |
| 商品编号 | commodity\_id | varchar | 不能 | 主键 |  |
| 商品名称 | commodity\_name | varchar | 不能 |  |  |
| 商品图片 | commodity\_picture | varchar | 不能 |  | 数据库只保存图片的地址，图片保存在硬盘里 |
| 商品价格 | commodity\_price | int | 不能 |  | 该价格只做参考，交易双方可私下协商 |
| 上架者帐号 | user\_id | varchar | 不能 | 外键 |  |
| 商品介绍 | commodity\_intro | varchar | 不能 |  |  |
| 商品关键词 | commodity\_key | varchar | 不能 |  | 关键词用于方便商品的查找 |
| 商品收藏量 | commodity\_scl | int | 不能 |  | 商品被用户收藏的数量，该数量作为放在主页的依据之一 |
| 商品评论量 | commodity\_pll | int | 不能 |  | 评论的数量，该数量作为放在主页的依据之一 |
| 商品上架时间 | commodity\_uptime | datetime | 不能 |  | 其他条件相同是，先上架的物品放在主页 |
| 商品上架积分 | commodity\_score | int | 不能 |  | 商品上架积分所需要的积分 |

### Order表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order | | | | | |
| 订单编号 | order\_id | varchar | 不能 | 主键 |  |
| 买家帐号 | user\_id1 | varchar | 不能 | 外键 |  |
| 卖家帐号 | user\_id2 | varchar | 不能 | 外键 |  |
| 交易地址 | order\_address | varchar | 不能 |  | 双方面对面交易的地址 |
| 买家手机号 | order\_tel1 | varchar | 不能 |  | 也可以是微信等联系方式 |
| 卖家手机号 | order\_tel2 | varchar | 不能 |  | 也可以是微信等联系方式 |
| 下单时间 | order\_time | datetime | 不能 |  | 买家在APP上购买的时间 |
| 交易额 | order\_cost | double | 不能 |  |  |
| 交易花费积分 | commodity\_score | int | 不能 |  |  |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |

### Message表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message | | | | | |
| 留言编号 | pl\_id | varchar | 不能 | 主键 |  |
| 留言时间 | pl\_time | datetime | 不能 |  |  |
| 留言内容 | pl\_content | varchar | 不能 |  | 内容上限为300个字符 |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |
| 发布者编号 | user\_id | varchar | 不能 | 外键 |  |

### Collection表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Collection | | | | | |
| 收藏编号 | sc\_id | varchar | 不能 | 主键 |  |
| 收藏时间 | sc\_time | datetime | 不能 |  |  |
| 收藏商品的价格 | sc\_price | varchar | 不能 |  |  |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |
| 收藏者编号 | user\_id | varchar | 不能 | 外键 |  |

# 四、命名规范

数据库命名使用匈牙利命名法

匈牙利命名法关键是：标识符的名字以一个或者多个小写字母开头作为前缀；前缀之后的是首字母大写的一个单词或多个单词组合，该单词要指明变量的用途。

编码阶段使用驼峰命名法

驼峰命名法，正如它的名称所表示的那样，指的是混合使用大小写字母来构成标识符的名字。其中第一个单词首字母小写，余下的单词首字母大写。

# 五、编码规则

## 5.1javascript书写位置

1、引入外部 js 文件

格式：<script src=”aaa.js”></script>

注：

html和jsp页面中不允许将后面那个标签去掉

ECMA第五版规定 type 属性不写时默认为 type=”text/javascript”，因此省略这个属性。

2.自己写js 代码

<script>

// 自定义的 js 代码

</script>

3.标签的位置

如果将 <script> 标签放到 <head> 标签中，等待外部js 文件全部下载、 解析并 执行完之后才能解析 <body> 中的元素，这会造成打开页面时会出 现短暂的吃顿现象。

因此我们要求将 <script> 标签放到 <body> 元素中，并且置于 <body> 元素中的 html 代码的最下面。

4.推荐

将 html 代码和 js 代码分离，也就是将 js 代码全部写到一个独立的 以 \*.js 的文 件中，再通过引入外部js 的方式引入到页面中。可以增加 代码的美观性和方便后期代 码维护。

## 5.2javascript命名规范

1.区分大小写

Javascript 中的一切（如：变量、函数名和操作符）都区分大小写。

如：

变量名 test 和变量名 Test 就是两个变量。

typeof 作为关键字不能作为变量名，但typeOf就可以作为变量名了， 因为它是一个新的变量。

2.标识符

标识符：变量、函数、属性和函数参数的名字。

推荐命名规则：

第一个字符必须是字母或下划线（\_）或美元符号（$）；

其它字符可以是字母或下划线或美元符号或数字；

用全英文命名的，采用驼峰大小写格式。即第一个字母小写，后面每 个有意义的首字母大写。

如：firstSecond myCar

标识符不能是关键字、保留字、true、false、null。

3.关键字和保留字

在使用中有不熟悉的关键字和保留字可查询下表：

关键字有：（javascript 中有意义的单词）ECMA第五版

关 键 字

break do instanceof typeof

case else new var

catch finally return void

continue for switch while

debugger function this with

default if throw try

delete in

保留字有：（javascript 中暂时没有意义，以后可能被用作关键字的单词） ECMA第三版

保 留 字

abstract enum（枚举） int short

boolean export interface static

byte extends long super

char final native synchronized

class float package throws

const goto private transient

double implements protected volatile

yield import public let

## 5.3注释

1.单行注释

// 单行注释

2.多行注释

/\*

\* 这是一个多行注释

\* 多行注释的第二行

\*/

3.方法注释

方法的注释一般应注明方法是干什么用的，注意事项是什么，参数含义， 返回数据信息。

/\*\*

\* 主要功能：生成随机数

\* 注意事项：此方法只能生成11-15位的随机数

\*

\* @param size 生成随机数的长度，范围在数字11~14之间

\* @return number 随机数

\*/

function getRandom (size) {

return Math.random().toString(16).substr(2, size);

}

## 5.4严格模式

'use strict';

严格模式 可以将js 代码中的一些不确定行为得到处理，对某些不 安全的操作也会抛出错误，强制性的将js 代码变得更加规范和不容易出 错。

上述看似字符串的xx 实际上是一个编译指示，用于告诉Javascript 引 擎切换到严格模式。

## 5.5模式

1.分号的使用

每一句代码的结尾推荐加上”;”，尽管在 javascript 中代码结束不加分号也不会报错。

加分号”;”的好处有：

避免错误，诸如不完整的输入；

可以通过删除多余的空格来压缩js 代码，代码结尾没有分号会导致压缩 错误；

增进代码性能。解析器不必花时间推测应该从哪里插入分号来结束语句。

2.代码块的使用

如：if条件语句中，如果代码块中只有一条语句

if (test)

alert(test);

上述代码不会导致错误，但不推荐使用。推荐使用的方式如下：

if (test) {

alert(test);

}

## 5.6编写规范

1.对于条件语句

if (条件) {

// 条件语句

}

注：if 与 () 和 {} 之间的空格

2.对于循环语句

for (var i = 0; i < 10; i++) {

// 循环语句

}

注：for的括弧内每个分号后面要空一格

3.对于方法

function test (param1, param2) {

// 方法体

}

注意：参数中逗号后面有个空格

4.js 代码中尽量使用单引号。

## 5.7变量

1.使用var 定义变量

javascript 里的变量是松散类型的，也就是即可以用 var 定义字符串类型的，也可以用var 定义整数类型的。

2. var message;

像这样只定义没有初始化值的，默认会保存一个特殊的值 —— undefined

3.同一个变量不推荐接收多种数据类型的值

var hhh = 'hello';

hhh = 111;

像这种首先用 hhh 变量接收字符串类型的值，再用 hhh 变量接收整型的值，语法上没有错误，但我们不推荐这样使用。

4.认识局部变量

function test () {

var message = 'hello';

}

test();

alert(message);

这里的message 是用 var 定义的局部变量，在调用方法test() 时被创建出来，在方法结束的时候也同时销毁了，所以alert(message) 会报错。

5.认识全局变量

function test () {

message = 'hello';

}

test();

alert(message);

这里的message 没有用var 定义，在调用方法 test() 时被创建出来， 没有用var 定义的变量属于全局变量，因此就算方法执行完毕也不会被回收 器回收，所有 alert(message) 时就会弹出 ‘hello’。

6.用 var 可以同时定义多个变量，中间用逗号”,”隔开。

如：

var aaa = 'hi',

bbb = 12,

ccc = true;

7.严格模式下注意事项

严格模式下变量名不能为 eval 和 arguments ，尽管这两个词既不是关 键字也不是保留字，但就是不能使用，否则报语法错误

# 用户模块代码

1. [↑](#footnote-ref-0)