

**《Web前端开发》课程项目**

****

基于Web前端技术的购物应用系统

**学生姓名**  卓如意

**学 号**  22145138

**院 系**  机械与电气工程学院

**班 级** 计算机科学与技术221

# 基于Web前端技术的购物应用系统

## 主页模块

### 任务要求

（1）界面元素：网络商城的名称、logo图片、分类菜单、客户姓名、购物车图标、购物车商品数量、版权信息（作者信息）。

（2）基本-商品列表：商品必须属性：Id，名称，简介，价格，库存数量，分类Id，图名称；商品可选属性：特点，产地，颜色，规格，尺寸，上架日期等；商品列表中商品数量在8-16之间；用户点击商品图片可以链接到商品详情页面；分页链接；

（3）选做-分类展示：点击分类名称，展示相应商品列表。（）（）

（4）选做-商品高亮：鼠标移动到商品上出现高亮效果，高亮的形式自定义。（）（）

（5）选做-二级分类：JS实现二级分类菜单，二级分类动态展示。（）（）

（6）选做-搜索功能：JS实现模糊搜索。（）（）

（7）选做-商品数据：JSon数据文件中读取显示商品信息。（）（）

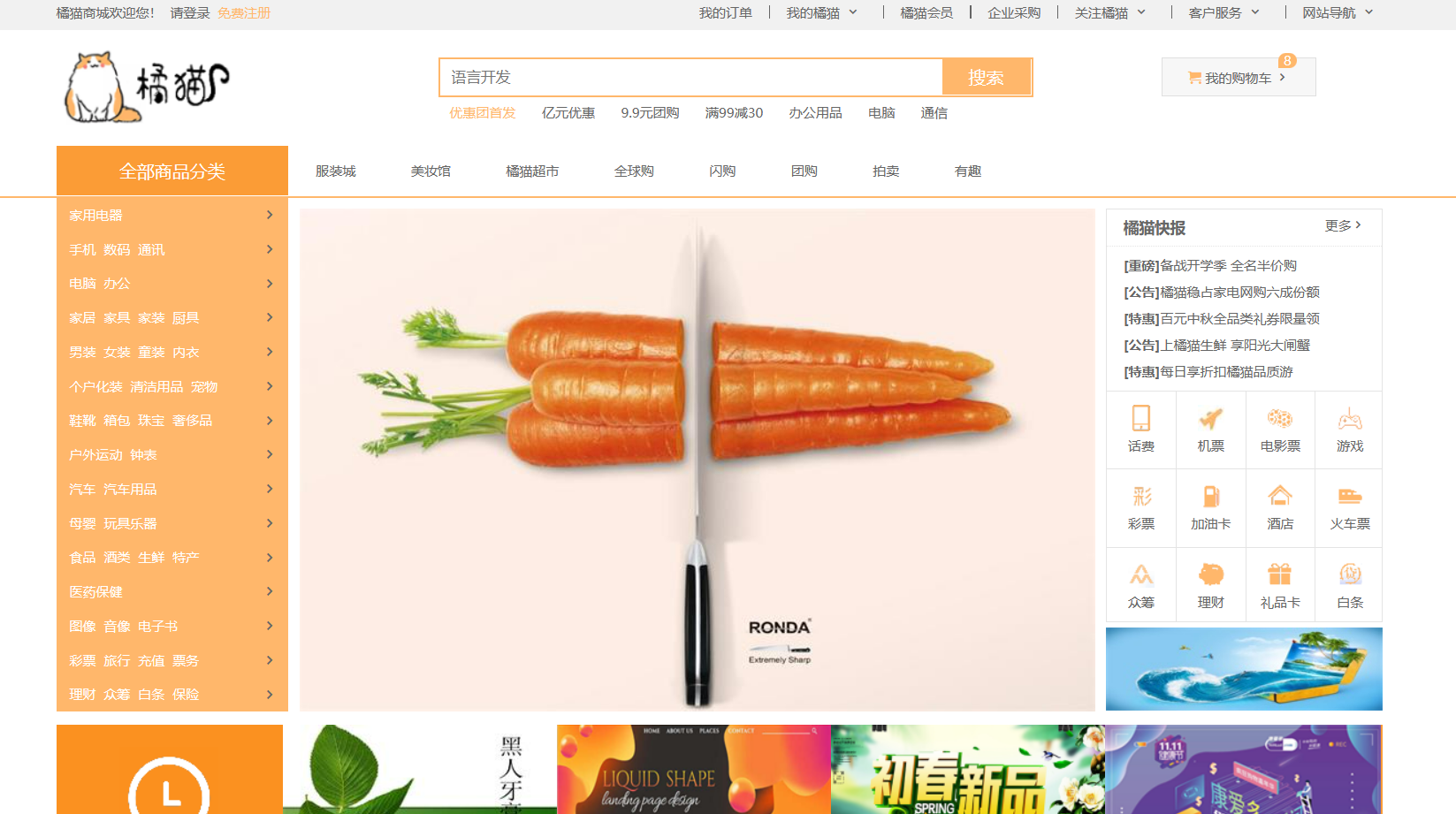
（8）选做-\*\*\*\*\*\*\*\***选做功能暂时不做，后续能做了再完善本页面**\*\*\*\*\*\*\*。

### 文件说明

表1‑1 主页模块文件说明

|  |  |  |
| --- | --- | --- |
| **文件目录** | **文件** | **说明** |
| / | index.html | 主页html文件 |
| /images | \*images\* | 页面修饰图片文件 |
| /pimg | \*images\* | 商品图片文件 |
| /styles | base.css | 通用基础css文件 |
| /styles | index.css | 主页css文件 |
| /scripts | index.js | 主页js文件 |
| /json |  | 商品json数据文件 |

### 页面效果



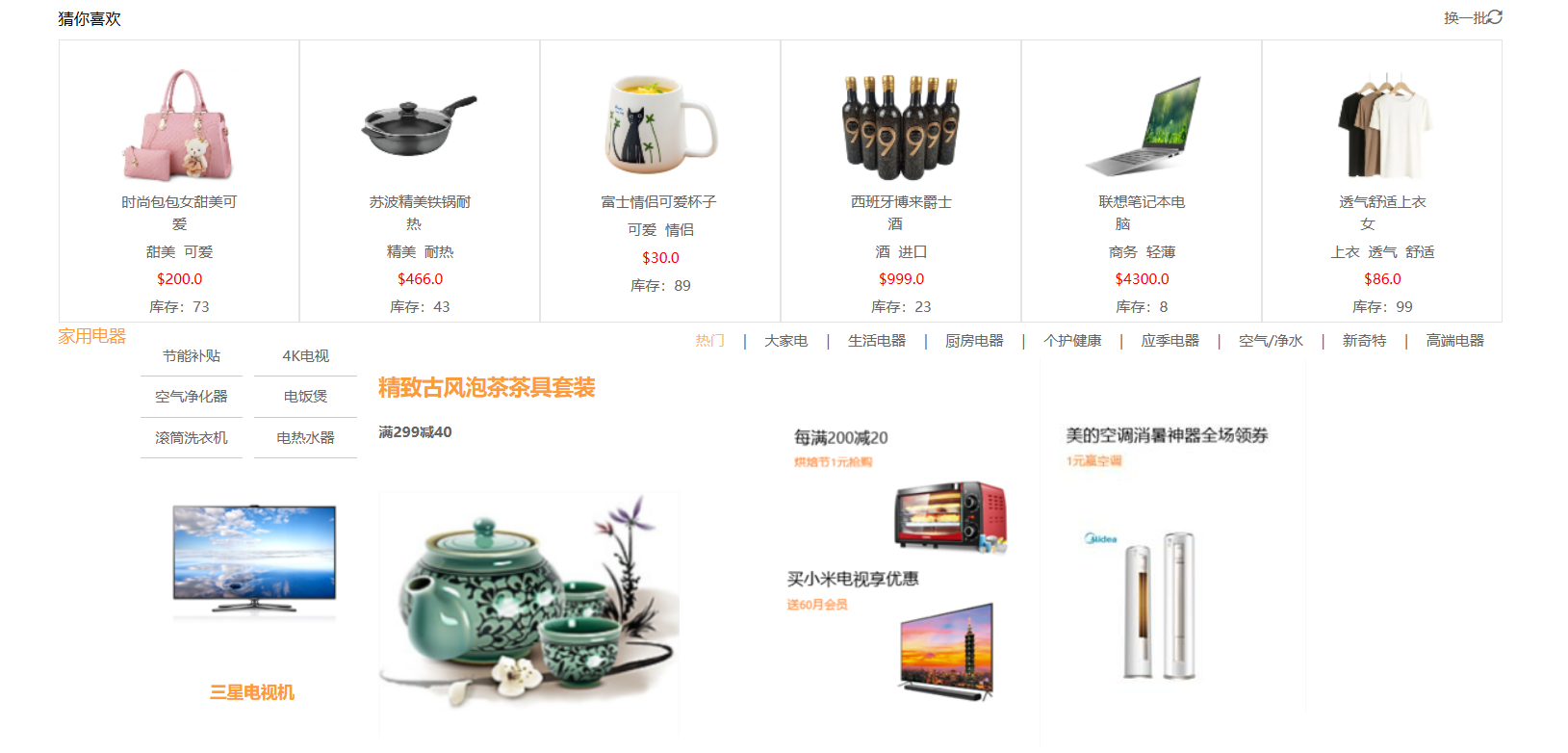


图1‑1 主页效果图

### HTML框架

























图1‑2 主页html框架图

### JS代码

/\* 无 \*/

## 详情模块

### 任务要求

（1）界面元素：商品详情图文、购物数量、加入购物车、继续购物

（2）基本-商品详情：商品详情展示；url参数读取；json数据读取

（3）基本-购物按钮：库存判断；加入购物车；信息提示

（4）选做-\*\*\*\*\*\*\*\*：\*\*\*\*\*\*\*\*\*其他选做功能可自行添加\*\*\*\*\*\*\*\*\*\*\*\*。（）

### 文件说明

表2‑1 详情模块文件说明

|  |  |  |
| --- | --- | --- |
| **文件目录** | **文件** | **说明** |
| / |  |  |
| /images | images |  |
| /pimg | pimg |  |
| /styles | main.css |  |
| /styles | reset.css |  |
| /scripts | Main.js |  |
| /json |  |  |

### 页面效果

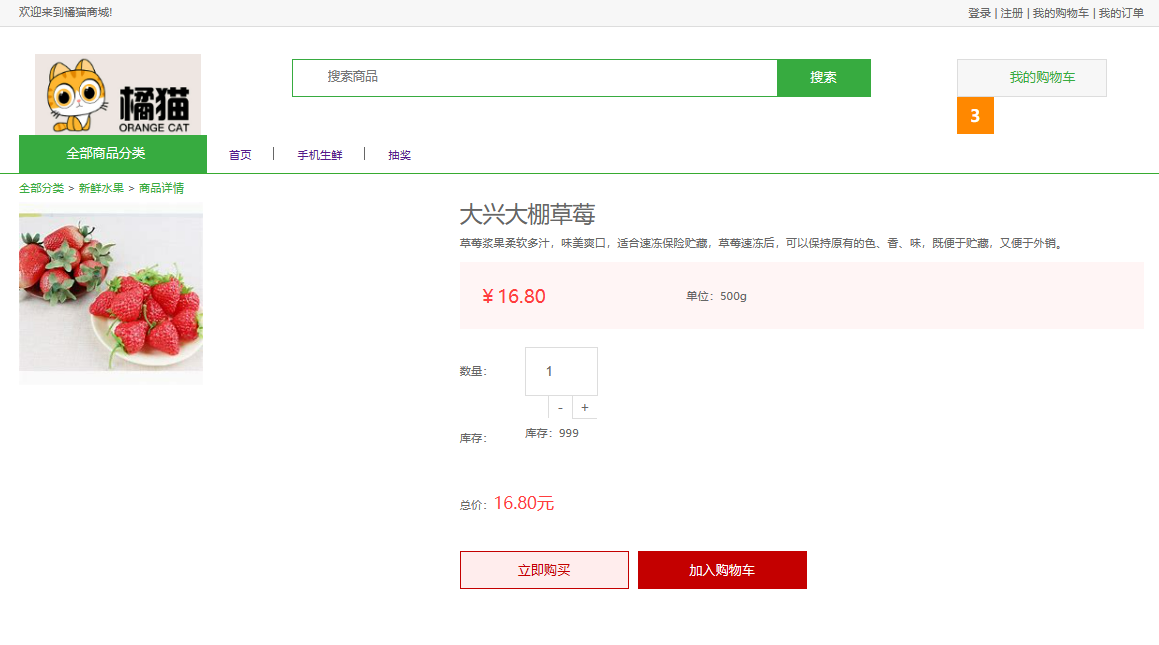


图2‑1 \*\*\*\*图

### HTML框架





图2‑2 详情html框架图

### JS代码

//实现前进和后退小图片的效果=====

let prev = document.querySelector('.prev');

let next = document.querySelector('.next');

let ul = document.querySelector('.spec-items ul');

prev.onclick = function () {

    ul.style.left = '-38px';

    prev.style.background = 'url(images/disabled-prev.png)';//以html文件为准

}

next.onclick = function () {

    ul.style.left = '-154px';

    next.style.background = 'url(images/disabled-next.png)';

}

/\*\*

 \* 可以有过渡效果：

 \* 1、数值类的

 \* 2、颜色类的

 \* 3、转换：平移、旋转、缩放、倾斜

 \* 盒子阴影

 \*/

//实现鼠标放在小图上，边框变红并且上面显示小图

let lis = document.querySelectorAll('.spec-items ul li');

let img = document.querySelector('.main-img  img');

let imgs = document.querySelectorAll('.spec-items img');

for (let i = 0; i < lis.length; i++) {

    lis[i].onmouseover = function () {

        for (let j = 0; j < lis.length; j++) {

            lis[j].className = 'null';

        }

        lis[i].className = 'img-hover';

        //两种表达：

        // 1、img.src=lis[i].children[0].src;

        //2、img.src=imgs[i].src;

        img.src = imgs[i].src;

    }

}

//实现鼠标滑入显示放大镜和大图的效果

let mainImg = document.querySelector('.main-img');//获取中图

let zoomPup = document.querySelector('.zoom-pup');//获取放大镜，中图上面的小黄块

let zoomDiv = document.querySelector('.zoom-div');//获取大图的div

let bigImg = document.querySelector('.zoom-div img')//获取大图

mainImg.onmouseover = function () {

    zoomPup.style.display = 'block';

    zoomDiv.style.display = 'block';

}

//实现鼠标滑出隐藏放大镜和大图的效果

mainImg.onmouseout = function () {

    zoomPup.style.display = 'none';

    zoomDiv.style.display = 'none';

}

//给中图绑定鼠标移动事件

mainImg.onmousemove = function (e) {

    //获取鼠标距离文档顶部(body)的距离pageX

    let pageY = e.pageY;

    //获取中图距离文档顶部(body)的距离  offsetTop

    let offsetTop = mainImg.offsetTop;

    //获取黄色小块的高度的一半

    let h = zoomPup.clientHeight / 2;

    //获取黄色小块顶部距离中图顶部的距离

    let top = pageY - offsetTop - h;

    //判断top

    if (top <= 0) {

        top = 0;

    } else if (top >= mainImg.clientHeight - zoomPup.clientHeight) {

        top = mainImg.clientHeight - zoomPup.clientHeight;

    }

    //将放大镜的定位top设置为黄色小块顶部距离中图顶部的距离。

    zoomPup.style.top = top + 'px';

    //获取鼠标距离文档左边(body)的距离 pageX

    let pageX = e.pageX;

    //获取中图距离文档左边(body)的距离

    let offsetLeft = mainImg.offsetLeft;

    //获取黄色小块宽度的一半

    let w = zoomPup.clientWidth / 2;

    //获取黄色小块距离中图左边的距离

    let left = pageX - offsetLeft - w;

    //判断left

    if (left <= 0) {

        left = 0;

    } else if (left >= mainImg.clientWidth - zoomPup.clientWidth) {

        left = mainImg.clientWidth - zoomPup.clientWidth;

    }

    //将放大镜的定位left设置为黄色小块顶部距离中图顶部的距离。

    zoomPup.style.left = left + 'px';

    let y = top / (mainImg.clientHeight - zoomPup.clientHeight);

    let yy = y \* (800 - 540);

    bigImg.style.top = -yy + 'px';//让大图向上走yy的距离

    let x = left / (mainImg.clientWidth - zoomPup.clientWidth);

    let xx = x \* (800 - 540);

    bigImg.style.left = -xx + 'px';//让大图向左走xx的距离

}

//实现购物车数量的改变

let reduce = document.querySelector('.reduce');

let add = document.querySelector('.add');

let buyNum = document.querySelector('.buy-num');

add.onclick = function () {

    buyNum.value++;

    if (buyNum.value > 1) {

        reduce.className = 'reduce';

    }

}

reduce.onclick = function () {

    if (buyNum.value > 1) {

        buyNum.value--;

    } else if (buyNum.value <= 1) {

        buyNum.value = 1;

        reduce.className = 'reduce disabled';

    }

}

//实现右边的图上下移动的效果

let shang = document.querySelector('.shang');

let xia = document.querySelector('.xia');

let ul1 = document.querySelector('.track-items ul');

shang.onclick = function () {

    ul1.style.top = '-369px';

}

xia.onclick = function () {

    ul1.style.top = '0px';

}

let lis1 = document.querySelectorAll('.track-items ul li');

for (let i = 0; i < lis.length; i++) {

    lis1[i].onmouseover = function () {

        for (let j = 0; j < lis1.length; j++) {

            lis1[j].className = 'null';

        }

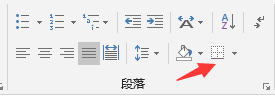
        lis1[i].className = 'img-flow';

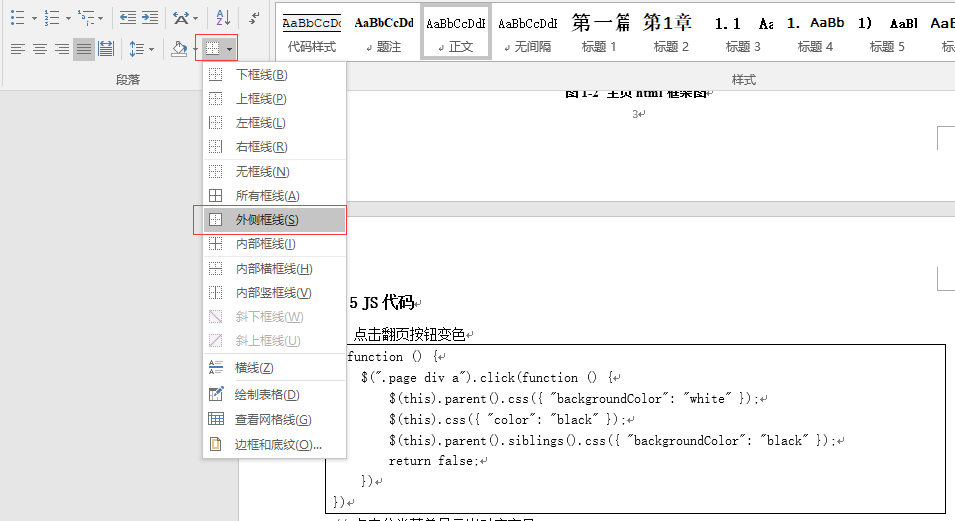
    }

}

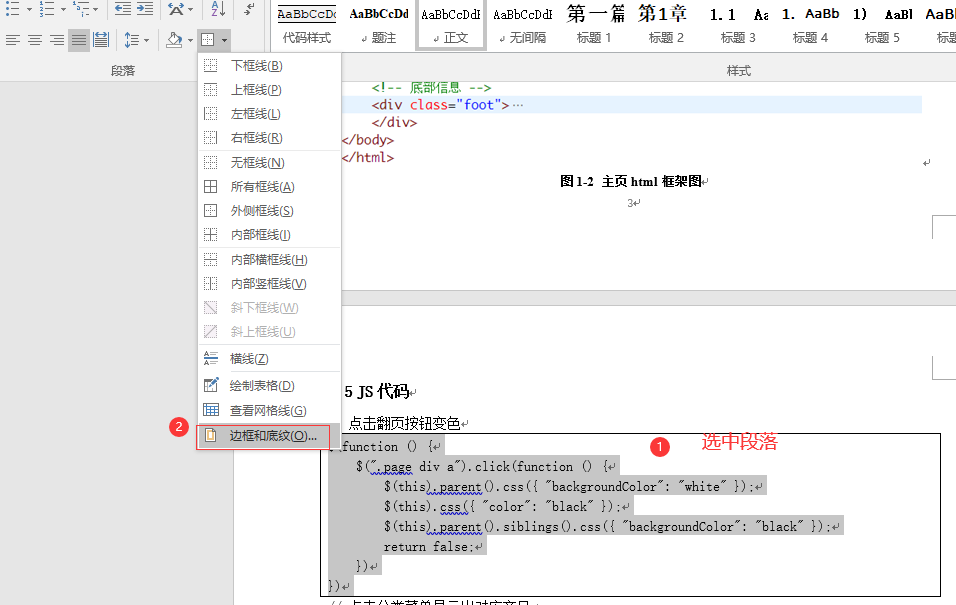
下面展示了如何设置代码在文档中的格式，供参考。

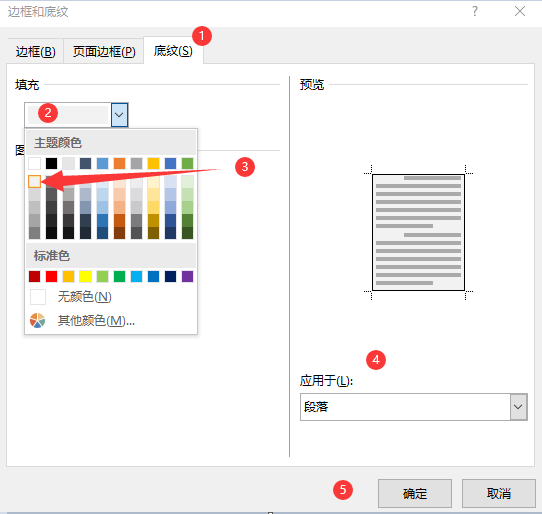
（格式1，边框设置）



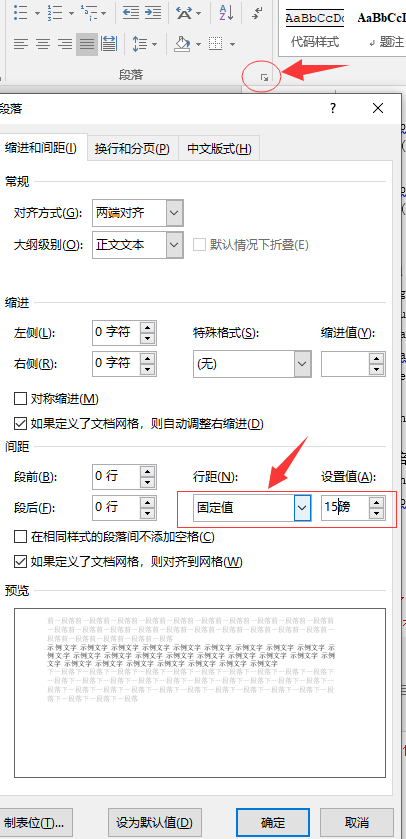


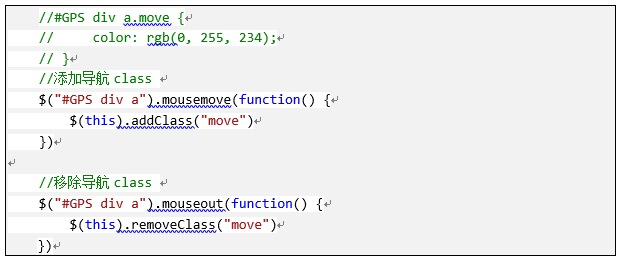
（格式2，底纹设置）





（格式3）代码段落的行距设为固定值15磅。





这种问题文字下面有白色底纹的修复方法如下：

