# AJAX知识点复盘

## 1.AJAX动态渲染页面

### 1.1获取图书信息渲染到页面:

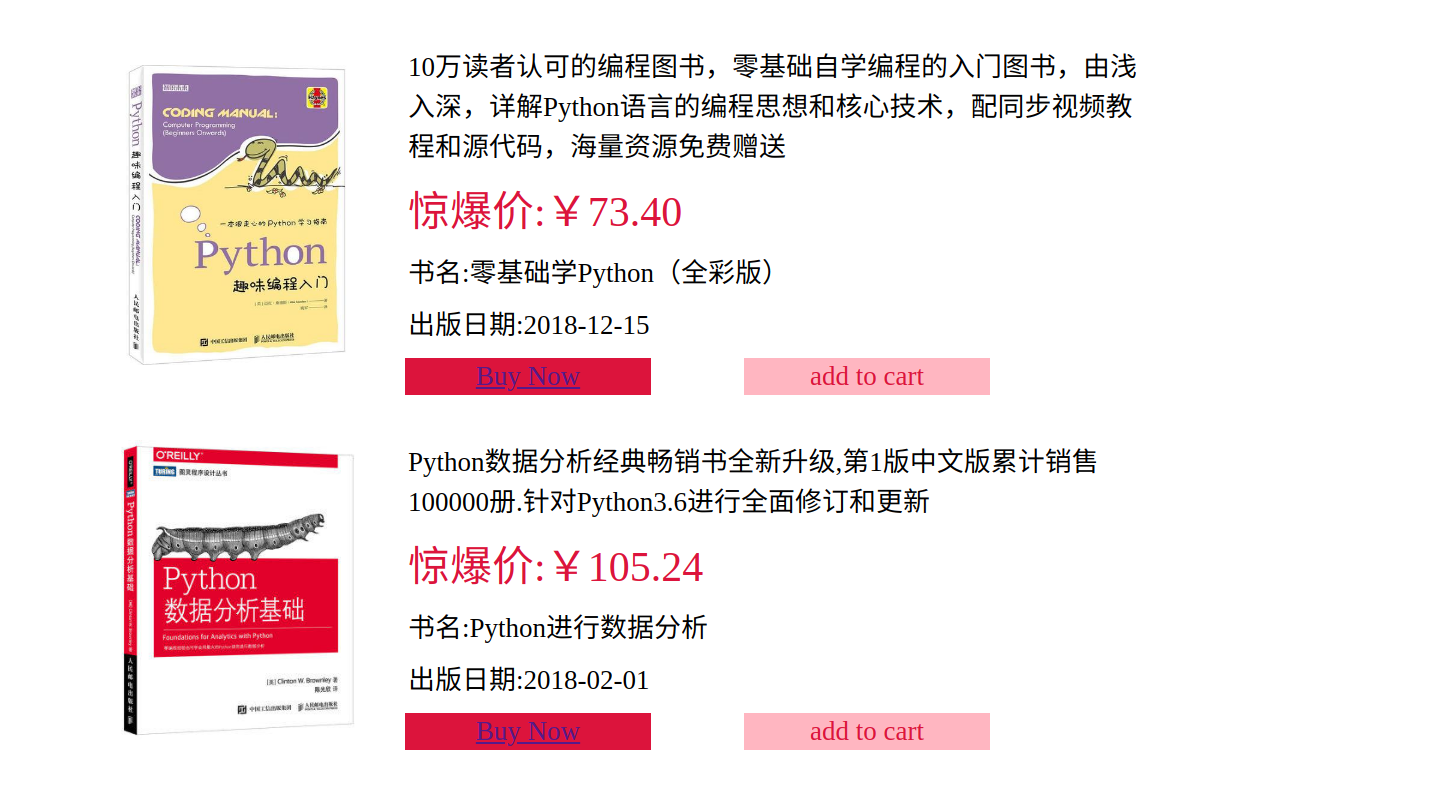
页面效果:

页面的

静态 两种图书信息相同的 页面效果

和

动态 不同的效果



### 1.2:分析页面需要动态渲染的标签:

* 分析页面标签结构
* 分析页面动态数据的位置

<table align="center" border="0" id="goods">  
 <tr>  
 <td>  
 <table border="0">  
 <tr>  
 <td><img src="/static/images/Lipstick.webp"></td>  
 <td valign="top">  
 <table border="0">  
 <tr>  
 <td width="250"><font color="black">Girl, you're going to be a woman</font></td>  
 </tr>  
 <tr valign="top">  
 <td ><font color="#dc143c" size="5">$16</font></td>  
 </tr>  
 <tr>  
 <td ><font color="black">Lipstick</font></td>  
 </tr>  
 <table border="0">  
 <tr>  
 <td bgcolor="#dc143c" align="center" width="80"><a href="http://product.dangdang.com/25160597.html" >Buy Now</a></td>  
 <td width="25"></td>  
 <td bgcolor="#ffb6c1" width="80" align="center"><font color="#dc143c">add to cart</font></td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <table border="0">  
 <tr>  
 <td><img src="/static/images/Lipstick.webp"></td>  
 <td valign="top">  
 <table border="0">  
 <tr>  
 <td width="250"><font color="black">Girl, you're going to be a woman</font></td>  
 </tr>  
 <tr valign="top">  
 <td ><font color="#dc143c" size="5">$16</font></td>  
 </tr>  
 <tr>  
 <td ><font color="black">Lipstick</font></td>  
 </tr>  
 <table border="0">  
 <tr>  
 <td bgcolor="#dc143c" align="center" width="80"><a href="http://product.dangdang.com/25160597.html" >Buy Now</a></td>  
 <td width="25"></td>  
 <td bgcolor="#ffb6c1" width="80" align="center"><font color="#dc143c">add to cart</font></td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <table border="0">  
 <tr>  
 <td><img src="/static/images/Lipstick.webp"></td>  
 <td valign="top">  
 <table border="0">  
 <tr>  
 <td width="250"><font color="black">Girl, you're going to be a woman</font></td>  
 </tr>  
 <tr valign="top">  
 <td ><font color="#dc143c" size="5">$16</font></td>  
 </tr>  
 <tr>  
 <td ><font color="black">Lipstick</font></td>  
 </tr>  
 <table border="0">  
 <tr>  
 <td bgcolor="#dc143c" align="center" width="80"><a href="http://product.dangdang.com/25160597.html" >Buy Now</a></td>  
 <td width="25"></td>  
 <td bgcolor="#ffb6c1" width="80" align="center"><font color="#dc143c">add to cart</font></td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
</table>

* 1.通过页面分析得到结果 页面展示的数据都在 表格标签#goods下的tr标签中，且通过tr标签的重复展示出网页内的数据
* 2.页面中包含的数据有 describe-商品描述,title-商品名字,picture-商品图片,price-商品价格,publisher\_date-出版日期.

### 1.3后端组织数据

* 后端组织相应数据结构交给前端渲染

books = Book.objects.all()  
 data\_list=[]  
 for book in books:  
 data={}  
 data['title'] = book.title  
 data['describe'] = book.describe  
 data['price'] = book.price  
 data['picture'] = 'http://127.0.0.1:8000/media/' + str(book.picture)  
 data['publisher\_date'] = book.publisher\_date  
 data\_list.append(data)  
 return JsonResponse({'code':200,'data\_list':data\_list})

### 1.4AJAX请求获取数据

* ajax动态渲染页面:

function shop(){  
 $.ajax({  
 type:'get',  
 url: 'http://127.0.0.1:8000/book',  
 datatype: 'json',  
 async: false,  
 success: function (response){  
 var data\_list = response.data\_list  
 // 获取到后端组织的数据结构  
 console.log(data\_list)  
 // 生成html代码  
 var html=''  
 for (var i=0;i<data\_list.length;i++){   
 html += '<tr><td><table border="0"><tr><td><img width="100" height="100" src="'  
 html += data\_list[i].picture  
 html += '"></td><td valign="top"><table border="0"><tr><td width="250"><font color="black">'  
 html += data\_list[i].describe  
 html += '</font></td></tr> <tr valign="top"><td ><font color="#dc143c" size="5">惊爆价:￥'   
 html += data\_list[i].price  
 html += '</font></td></tr><tr><td ><font color="black">书名:'  
 html += data\_list[i].title  
 html += '</font></td></tr>'  
 html += '<tr><td ><font color="black">出版日期:'  
 html += data\_list[i].publisher\_date  
 html += '</font></td></tr><table border="0"><tr><td bgcolor="#dc143c" align="center" width="80"><a href="" >Buy Now</a></td><td width="25"></td>'  
 html += '<td bgcolor="#ffb6c1" width="80" align="center"><font color="#dc143c">add to cart</font></td></tr></table></td></tr></table></td></tr>'  
 }  
 // 向网页中写入html代码  
 $('#goods').html(html)  
 }   
 })  
}  
// 页面加载完成，调用shop()进行动态数据渲染  
window.onload = function () {  
 shop()  
}

## 2.ajax-post请求表单提交及后端取值办法

### 2.1前端传输数据

* 知识点：
* csrf跨站点攻击解决方案1-将{%csrf\_token%}标签值写入到post提交的数据中

$.ajax({  
 type:'POST',  
 url:'http://127.0.0.1:8000/book/',  
 datatype:JSON,  
 data:{  
 'username':'mitono',  
 'addr':'北京',  
 // {%csrf\_token%}标签提交方式1:放到post提交的数据中键为csrfmiddlewaretoken  
 'csrfmiddlewaretoken':$('[name="csrfmiddlewaretoken"]').val()  
 },  
 async: false,  
 success: function (response){  
 console.log(response)  
 }  
 })

### 2.2后端获取提交的数据

if request.method == "POST":  
 # 普通提交取值办法  
 print(request.POST.get('username'))  
 print(request.POST.get('addr'))  
 return HttpResponse('ok')

## 3.ajax-post请求JSON提交及后端取值办法.

### 3.1前端传输数据

* 知识点
* csrf跨站点攻击解决方案2-post提交中设置请求一个请求头X-csrftoken
* contentType的设置

$.ajax({  
 type:'POST',  
 url:'http://127.0.0.1:8000/book/',  
 datatype:JSON,  
 contentType: "application/json;charset=utf-8",  
 // 请求头提交方式.  
 headers:{"X-csrftoken":$("[name='csrfmiddlewaretoken']").val()},  
 data:JSON.stringify({  
 'username':'mitono',  
 'addr':'北京',  
 }),  
 async: false,  
 success: function (response){  
 console.log(response)  
 }  
 })

### 3.2后端传输数据

if request.method == "POST":  
 # json.string()提交取值办法  
 print(json.loads(request.body).get("username"))  
 print(json.loads(request.body).get("addr"))  
 return HttpResponse('ok')

### 3.3说明:

1.contentType设置为"application/json;charset=utf-8",数据一定要使用JSON.stringify()转成json字符串再提交,否则后端使用json.loads(request.body)取值会报json.decoder.JSONDecodeError错误.

2.JSON.string()提交数据,一定要设置contentType为application/json;charset=utf-8,否则ajax的提交方式还是表单提交.

3.{%csrf\_token%}标签本质是一个name=csrfmiddlewaretoken的input标签.