**1 简述对Ajax的理解**

**参考答案**

AJAX是Asynchronous JavaScript and Xml 异步的JavaScript和Xml。它一种用来改善用户体验的技术，其实质是，使用XMLHttpRequest对象异步地向服务器发请求。服务器返回部分数据，而不是一个完整的页面，以页面无刷新的效果更改页面中的局部内容。

**2 下列说法正确的是（）**

A. Ajax的核心对象是XMLHttpResponse对象

B. Ajax的XMLHttpRequest对象的status属性代表响应的状态码

C. 编写Ajax程序时，无需对浏览器进行判断即可创建XMLHttpRequest对象

D. 编写Ajax程序时，需要为readystatechange事件添加回调处理

**参考答案**

本题的正确答案为BD。

A选项说法错误。XMLHttpRequest是Ajax的核心对象。

C选项说法错误。编写Ajax程序时，需要对浏览器进行判断后，再创建XMLHttpRequest对象。

**3 简述使用AJAX进行编程的步骤**

**参考答案**

使用AJAX编程的步骤如下：

1. 获取AJAX对象：获取 XMLHttpRequest对象。

2. 创建请求：调用XMLHttpRequest对象的open方法。

3. 设置回调函数：为AJAX对象的 onreadystatechange事件设定响应函数。

4. 发送请求：调用Ajax对象的send方法。

**4 使用Ajax检验注册信息中的昵称是否存在**

**参考答案**

实现此案例，可以按照如下步骤进行

**步骤一：新建regist.html文件**

新建regist.html文件，页面代码如图-1所示



图-1

**步骤二：使用js编写oncheck\_name()方法**

在regist.html中，编写js代码，使用AJAX验证昵称是否存在，代码如图-2所示

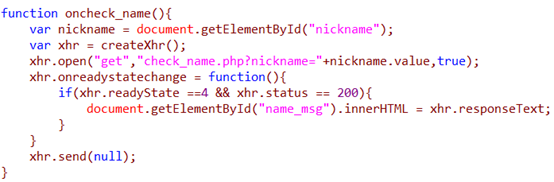


图-2

**步骤三：创建check\_name.php**

创建check\_name.php，在该文件中主要判断昵称是否存在。代码如图-3所示



图-3

**步骤四：运行，查看结果**

当输入存在的昵称时，如下图-4所示

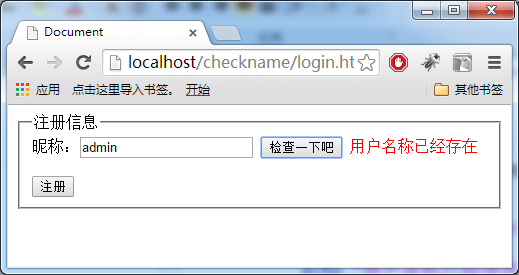


图-4

输入昵称不存在时，效果如下图所示：

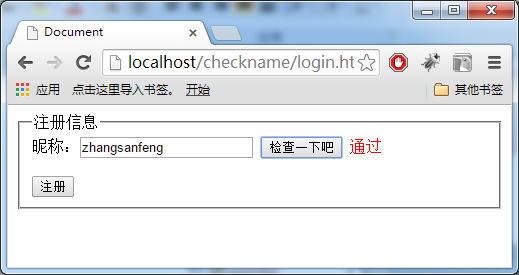


图-5

本案例的完整代码如下。

regist.html文件的完整代码如下：

1. **<!**doctype html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta charset**=**"UTF-8"**>**
5. **<**title**>**Document**</**title**>**
6. **<**script**>**
7. **function** createXhr**(){**
8. **if(**window**.**XMLHttpRequest**){**
9. **return** **new** XMLHttpRequest**();**
10. **}else{**
11. **return** **new** ActiveXObject**(**"Microsoft.XMLHTTP"**);**
12. **}**
13. **}**
14. **function** oncheck\_name**(){**
15. **var** nickname **=** document**.**getElementById**(**"nickname"**);**
16. **var** xhr **=** createXhr**();**
17. xhr**.**open**(**"get"**,**"check\_name.php?nickname="**+**nickname**.**value**,true);**
18. xhr**.**onreadystatechange **=** **function(){**
19. **if(**xhr**.**readyState **==4** **&&** xhr**.**status **==** **200){**
20. document**.**getElementById**(**"name\_msg"**).**innerHTML **=** xhr**.**responseText**;**
21. **}**
22. **}**
23. xhr**.**send**(null);**
24. **}**
25. **</**script**>**
26. **</**head**>**
27. **<**body**>**
28. **<**fieldset**>**
29. **<**legend**>**注册信息**</**legend**>**
30. 昵称：**<**input name**=**"nickname" id**=**"nickname"**/>**
31. **<**input type**=**"button" value**=**"检查一下吧" onclick**=**"oncheck\_name()"**/>**
32. **<**span id**=**"name\_msg" style**=**"color:red;"**></**span**>**
33. **<**br**/><**br**/>**
34. **<**input type**=**"submit" value**=**"注册"**/>**
35. **</**fieldset**>**
36. **</**body**>**
37. **</**html**>**

check\_name.php文件完整代码。

1. **<?**php
2. $nickname **=** $\_REQUEST**[**"nickname"**];**
3. **if(**$nickname **==** "admin"**){**
4. echo "用户名称已经存在"**;**
5. **}else{**
6. echo "通过"**;**
7. **}**
8. **?>**

**1 下列XML文档中，符合XML语法规范的是**

A．

1. **<**customer**>**
2. **<**address**>123** MainStreet**></**Address**>**
3. **</**customer**>**

B．

1. **<**customer**>**
2. **<**name**>**Joe’s XML Works**</**name**>**
3. **<**address**>**New York
4. **</**costomer**>**

C．

1. **<**customer type**=**extemal**>**
2. **<**name**>**Partners Unlimited**</**name**>**
3. **</**customer**>**

D.

1. **<**customer name**=**"JohnDoe"**>**
2. **<**address**>123** Main Street**</**address**>**
3. **<**zip code**=**"01837"**/>**
4. **</**customer**>**

**参考答案**

本题正确答案为D。

AB选项，不符合XML语法规范。在XML中，元素要成对出现，而且大小写敏感。A选项元素address的结尾为</Address>，“A”字母要小写。B选项元素<address>没有结尾标记。

C选项，不符合XML语法规范。在XML中，属性必须被引号包围，不过单引号和双引号均可使用。

**2 通过给定的XML文件内容构建如图-1所示的网页程序**

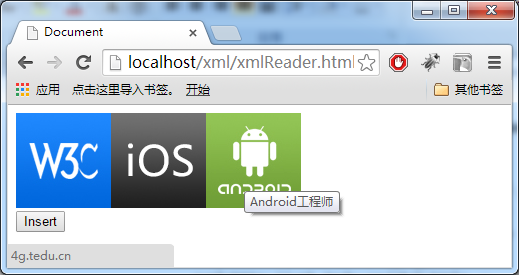


图-1

给定的XML文件内容如下：

<?xml version="1.0" encoding="UTF-8" ?>

<root>

<imags>

<imgsrc>images/webfront.jpg</imgsrc>

<title>WEB 前端工程师</title>

<url>http://web.tedu.cn/zt/</url>

</imags>

<imags>

<imgsrc>images/ios.jpg</imgsrc>

<title>IOS工程师</title>

<url>http://4g.tedu.cn/</url>

</imags>

<imags>

<imgsrc>images/android.gif</imgsrc>

<title>Android工程师</title>

<url>http://4g.tedu.cn/</url>

</imags>

</root>

其中，imgsrc表示图像的存储路径，title表示鼠标移入到图像时所显示的文本信息，url表示点击时跳转的路径。当点击Insert按钮时能够显示以上三幅图像以及各个功能。

**参考答案**

实现此案例需要按照如下步骤进行

**步骤一：创建xmlReader.html页面**

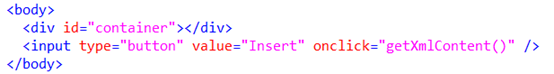


图-2

**步骤二：创建createXhr方法**

创建createXhr方法，用于根据不同的浏览器创建不同的xhr对象，代码如图-3所示



图-3

**步骤三：创建getXmlContent方法，读取xml数据**

创建getXmlContent方法，通过xhr加载和读取xml文档中的内容。并获取到所有的imgs的节点，代码如图-4所示

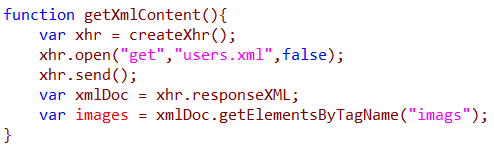


图-4

**步骤四：循环遍历images的所有子节点，并获取数据**

通过循环来获取到images中的所有子节点，并且根据节点数量创建a标签以及img标签。再遍历已经获取出来的节点，用户得到具体的节点信息，需要通过nodeType属性判断是否为元素节点（即nodeType == 1）。将数据添加到对应的节点中，再将节点添加到container中。详细代码如图-5所示。



图-5

本案例中完整代码如下：

users.xml 完整代码：

1. **<?**xml version**=**"1.0" encoding**=**"UTF-8" **?>**
2. **<**root**>**
3. **<**imags**>**
4. **<**imgsrc**>**images**/**webfront**.**jpg**</**imgsrc**>**
5. **<**title**>**WEB 前端工程师**</**title**>**
6. **<**url**>**http**:**//web.tedu.cn/zt/</url>
7. **</**imags**>**
8. **<**imags**>**
9. **<**imgsrc**>**images**/**ios**.**jpg**</**imgsrc**>**
10. **<**title**>**IOS工程师**</**title**>**
11. **<**url**>**http**:**//4g.tedu.cn/</url>
12. **</**imags**>**
13. **<**imags**>**
14. **<**imgsrc**>**images**/**android**.**gif**</**imgsrc**>**
15. **<**title**>**Android工程师**</**title**>**
16. **<**url**>**http**:**//4g.tedu.cn/</url>
17. **</**imags**>**
18. **</**root**>**

xmlReader.html完整代码：

1. **<!**doctype html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta charset**=**"UTF-8"**>**
5. **<**title**>**Document**</**title**>**
6. **<**script**>**
7. **function** createXhr**(){**
8. **var** xhr**=null;**
9. **if(**window**.**XMLHttpRequest**){**
10. xhr**=new** XMLHttpRequest**();**
11. **}else{**
12. xhr**=new** ActiveXObject**(**"Microsoft.XMLHTTP"**);**
13. **}**
14. **return** xhr**;**
15. **}**
16. **function** getXmlContent**(){**
17. **var** xhr **=** createXhr**();**
18. xhr**.**open**(**"get"**,**"users.xml"**,false);**
19. xhr**.**send**();**
20. **var** xmlDoc **=** xhr**.**responseXML**;**
21. **var** images **=** xmlDoc**.**getElementsByTagName**(**"imags"**);**
22. **for(var** i **=** **0** **;** i **<** images**.**length **;** i **++){**
23. **var** image **=** images**[**i**];**
24. **var** a **=** document**.**createElement**(**"a"**);**
25. **var** img **=** **new** Image**();**
26. **for(var** j**=0;**j **<** image**.**childNodes**.**length **;** j **++){**
27. **var** node **=** image**.**childNodes**[**j**];**
28. **if(**node**.**nodeType **==** **1){**
29. **if(**node**.**nodeName **==** "imgsrc"**){**
30. img**.**src **=** node**.**childNodes**[0].**nodeValue**;**
31. **}**
32. **if(**node**.**nodeName **==** "title"**){**
33. img**.**title **=** node**.**childNodes**[0].**nodeValue**;**
34. **}**
35. **if(**node**.**nodeName **==** "url"**){**
36. a**.**setAttribute**(**"href"**,**node**.**childNodes**[0].**nodeValue**);**
37. **}**
38. a**.**appendChild**(**img**);**
39. document**.**getElementById**(**"container"**).**appendChild**(**a**);**
40. **}**
41. **}**
42. **}**
43. **}**
44. **</**script**>**
45. **</**head**>**
46. **<**body**>**
47. **<**div id**=**"container"**></**div**>**
48. **<**input type**=**"button" value**=**"Insert" onclick**=**"getXmlContent()" **/>**
49. **</**body**>**
50. **</**html**>**

**3 JSON对象的设计和解析**

需求描述：网上书城项目中，书籍展示页面需要列举出如下图-6所示的书籍信息，请设计合适的数据结构描述这些数据。

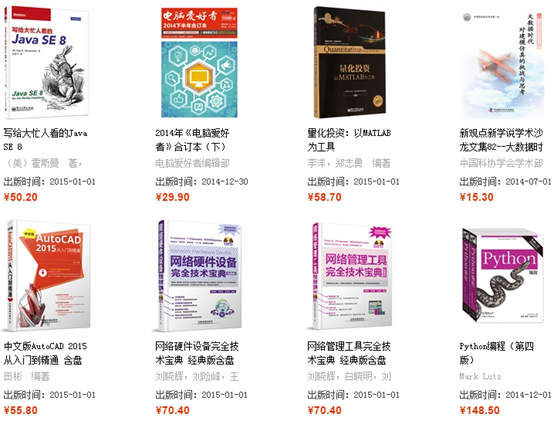


图-6

**参考答案**

根据需求描述及设计图样，数据整体是一个书籍对象的数组，每个书籍对象又需要图片(在页面中是img的存储路径)、书名、作者、出版时间、价格等成员属性；此外，当鼠标放到每个书籍图片上时，超链接地址中还包含书籍的编号属性。故可以使用JSON格式存储这些数据。

具体步骤如下所示：

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为exercise07\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**JSON对象的设计和解析**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加JavaScript和CSS文件引用，以及提示文字**

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**JSON对象的设计和解析**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise07\_02.js"**></script>**
7. **<link rel=**"stylesheet" **href=**"exercise07\_02.css" **type=**"text/css"**/>**
8. **</**head**>**
9. **<**body**>**
10. **<p>网上书城-商品列表</p>**
11. **<hr/>**
12. **</**body**>**
13. **</**html**>**

向页面中添加测试数据，先预览下效果。

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**JSON对象的设计和解析**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise07\_02.js"**></**script**>**
7. **<**link rel**=**"stylesheet" href**=**"exercise07\_02.css" type**=**"text/css"**/>**
8. **</**head**>**
9. **<**body**>**
10. **<**p**>**网上书城**-**商品列表**</**p**>**
11. **<**hr**/>**
12. **<div id=**"container"**>**
13. **<div class=**"book"**>**
14. **<img src=**"img/b1001.jpg"**/>**
15. **<a href=**"bookdetail.do?bno=1001"**>书名</a>**
16. **<span class=**"author"**>作者</span>**
17. **<span class=**"price"**>￥99</span>**
18. **</div>**
19. **</div>**
20. **</**body**>**
21. **</**html**>**

**步骤三：创建外部exercise07\_02.css文件，调整页面显示效果**

1. #container**{**
2. width**:** 600px**;**
3. **}**
4. **.**book**{**
5. width**:** 150px**;**
6. float**:** left**;**
7. font**-**size**:** **0.**8em**;**
8. **}**
9. **.**book a**{**
10. color**:** #**000000;**
11. text**-**decoration**:** none**;**
12. **}**
13. **.**book **.**author**{**
14. color**:** #aaaaaa**;**
15. display**:** block**;**
16. **}**
17. **.**book **.**price**{**
18. color**:** #ff0000**;**
19. display**:** block**;**
20. **}**

**步骤四：创建外部exercise07\_02.js文件，使用JSON表达式描述书籍数组**

1. //使用JSON格式存储页面数据
2. **var** bookList **=** **[** **{**
3. "bno" **:** **1001,**
4. "bname" **:** '写给大忙人看的Java SE 8'**,**
5. "img" **:** 'img/b1001.jpg'**,**
6. "author" **:** '（美）霍斯曼　著'**,**
7. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
8. "price" **:** **50.2**
9. **},** **{**
10. "bno" **:** **1002,**
11. "bname" **:** '2014年《电脑爱好者》合订本（下）'**,**
12. "img" **:** 'img/b1002.jpg'**,**
13. "author" **:** '电脑爱好者编辑部'**,**
14. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
15. "price" **:** **29.9**
16. **},** **{**
17. "bno" **:** **1003,**
18. "bname" **:** '量化投资：以MATLAB为工具'**,**
19. "img" **:** 'img/b1003.jpg'**,**
20. "author" **:** '李洋，郑志勇　编著'**,**
21. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
22. "price" **:** **58.7**
23. **},** **{**
24. "bno" **:** **1004,**
25. "bname" **:** '新观点新学说学术沙龙文集'**,**
26. "img" **:** 'img/b1004.jpg'**,**
27. "author" **:** '中国科协学会学术部'**,**
28. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
29. "price" **:** **15.3**
30. **},** **{**
31. "bno" **:** **1005,**
32. "bname" **:** '中文版AutoCAD 2015从入门到精通'**,**
33. "img" **:** 'img/b1005.jpg'**,**
34. "author" **:** '田彬　编著'**,**
35. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
36. "price" **:** **55.8**
37. **},** **{**
38. "bno" **:** **1006,**
39. "bname" **:** '网络硬件设备完全技术宝典'**,**
40. "img" **:** 'img/b1006.jpg'**,**
41. "author" **:** '刘晓辉，刘险峰，王'**,**
42. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
43. "price" **:** **70.4**
44. **},** **{**
45. "bno" **:** **1007,**
46. "bname" **:** '网络管理工具完全技术宝典'**,**
47. "img" **:** 'img/b1007.jpg'**,**
48. "author" **:** '刘晓辉，白晓明，刘'**,**
49. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
50. "price" **:** **70.4**
51. **},** **{**
52. "bno" **:** **1008,**
53. "bname" **:** 'Python编程（第四版）'**,**
54. "img" **:** 'img/b1008.jpg'**,**
55. "author" **:** 'Mark Lutz'**,**
56. "pubDate" **:** **new** Date**(2014,** **10,** **25),**
57. "price" **:** **148.5**
58. **}** **];**

**步骤五：修改exercise07\_02.html，使用JavaScript解析JSON数据，输出为页面元素**

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**JSON对象的设计和解析**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise07\_02.js"**></**script**>**
7. **<**link rel**=**"stylesheet" href**=**"exercise07\_02.css" type**=**"text/css"**/>**
8. **</**head**>**
9. **<**body**>**
10. **<**p**>**网上书城**-**商品列表**</**p**>**
11. **<**hr**/>**
12. **<**div id**=**"container"**>**
13. **<script type=**"text/javascript"**>**
14. **for(var i=0; i<bookList.length; i++){**
15. **var b = bookList[i];**
16. **with(b){**/\*with代码块中，所有的变量都是b的属性，如img相当于b.img\*/
17. **document.**write**(**'<div class="book"><img src="'**+img+**'"/><a href="bookdetail.do?bno='**+bno+**'">'**+bname+**'</a><span class="author">'**+author+**'</span><span class="price">￥'**+price+**'</span></div>'**);**
18. **}**
19. **}**
20. **</script>**
21. **</**div**>**
22. **</**body**>**
23. **</**html**>**

**1 编写程序，实现类似于百度搜索时的自动完成功能**

使用jQuery + AJAX实现类似于百度搜索时自动完成功能，界面如图-1所示

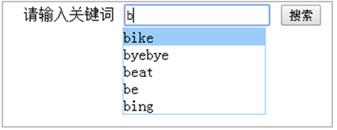


图-1

**参考答案**

首先在输入框上注册keyup事件，然后在事件中通过AJAX获取JSON对象。取得数据后，每一项数据创建一个li标签，在标签上注册click事件，这样当我们点击每一项的时候，就可以响应事件。键盘导航的关键是记录当前高亮的索引值，根据索引值来调整背景高亮。显示下拉列表的位置要根据输入框的位置来设置，当浏览器的大小改变时，随时调整下拉列表的位置。

完成本案例可按照如下步骤进行。

**步骤一：新建baidu.html文件**

新建baidu.html文件，在该文件中创建如图-1所示的界面，代码如图-2所示。

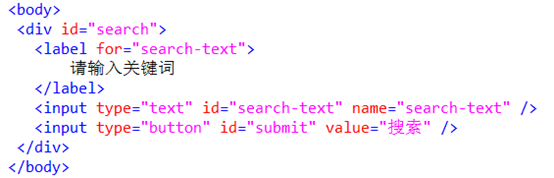


图-2

**步骤二：引入jQuery.js文件**

在baidu.html文件中，引入jQuery.js文件，代码如图-3所示。

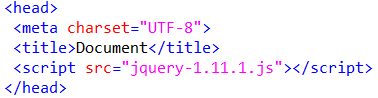


图-3

**步骤三：设置页面样式**

在baidu.html中，添加页面样式信息，代码如图-4所示。

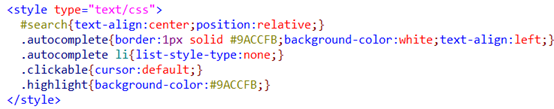


图-4

**步骤四：创建getResponse.php**

创建getResponse.php，在里面模拟服务器提供的数据，代码如图-5所示



图-5

**步骤五：获取jQuery对象**

获取id为"search"的div层对应的jQuery对象以及id为"search-text"的文本框对应的jQuery对象，代码如图-6所示。

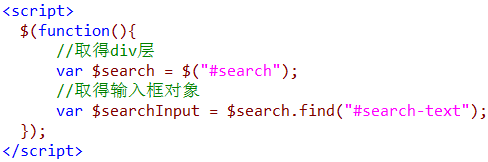


图-6

**步骤六：创建自动完成的下拉列表**

首先，关闭浏览器对输入框的自动完成功能；然后创建自动完成的列表，用于显示服务器返回的数据，插入在搜索按钮的后面，待显示的时候再调整位置，代码如图-7所示。

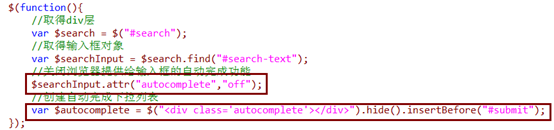


图-7

**步骤七：创建clear方法并给输入框注册blur事件**

首先，创建clear方法，用于清空下拉列表的内容并且隐藏下拉列表区；然后给输入框注册blur事件，当输入框失去焦点的时候清空下拉列表并隐藏，代码如图-8所示。

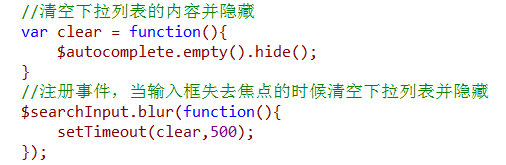


图-8

**步骤八：设置下拉项的背景高亮**

新建方法setSelectedItem，该方法用于设置下拉项的高亮背景，代码如图-9所示。



图-9

**步骤九：发送Ajax请求**

新建方法ajax\_request，在该方法中发送AJAX请求，通过AJAX请求获取JSON对象。取得数据后，每一项数据创建一个li标签，在标签上注册click事件，这样当点击每一项的时候，就可以响应事件，代码如图-10所示。





图-10

**步骤十：在输入框上注册keyup事件**

在输入框上注册keyup事件，然后在事件中通过AJAX获取JSON对象。取得数据后，每一项数据创建一个li标签，在标签上注册click事件，这样当点击每一项的时候，就可以响应事件。键盘导航的关键是记录当前高亮的索引值，根据索引值来调整背景高亮，代码如图-11，图-12所示。



图-11



图-12

**步骤十一：调整下拉列表框的位置**

注册窗口大小改变的事件，重新调整下拉列表的位置，代码如图-13所示。

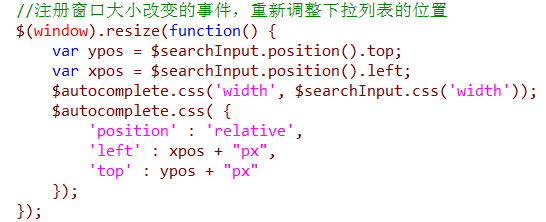


图-13

本案例完整代码如下。

baidu.html完整代码如下：

1. **<!**doctype html**>**
2. **<**html lang**=**"en"**>**
3. **<**head**>**
4. **<**meta charset**=**"UTF-8"**>**
5. **<**title**>**搜索关键词**</**title**>**
6. **<**style type**=**"text/css"**>**
7. #search**{**text**-**align**:**center**;**position**:**relative**;}**
8. **.**autocomplete**{**border**:**1px solid #9ACCFB**;**background**-**color**:**white**;**text**-**align**:**left**;}**
9. **.**autocomplete li**{**list**-**style**-**type**:**none**;}**
10. **.**clickable**{**cursor**:default;}**
11. **.**highlight**{**background**-**color**:**#9ACCFB**;}**
12. **</**style**>**
13. **<**scriptsrc**=**"jquery-1.11.1.js"**></**script**>**
14. **<**script**>**
15. $**(function(){**
16. //取得div层
17. **var** $search **=** $**(**"#search"**);**
18. //取得输入框对象
19. **var** $searchInput **=** $search**.**find**(**"#search-text"**);**
20. //关闭浏览器提供给输入框的自动完成功能
21. $searchInput**.**attr**(**"autocomplete"**,**"off"**);**
22. //创建自动完成下拉列表
23. **var** $autocomplete **=** $**(**"<div class='autocomplete'></div>"**).**hide**().**insertBefore**(**"#submit"**);**
24. //清空下拉列表的内容并隐藏
25. **var** clear **=** **function(){**
26. $autocomplete**.**empty**().**hide**();**
27. **}**
28. //注册事件，当输入框失去焦点的时候清空下拉列表并隐藏
29. $searchInput**.**blur**(function(){**
30. setTimeout**(**clear**,500);**
31. **});**
32. //当鼠标或者键盘的上下键移动高亮项目，像百度一样
33. varselectedItem **=** **null;**
34. vartimeoutid **=** **null;**
35. //设置下拉项的背景高亮
36. varsetSelectedItem **=** **function(**item**){**
37. //更新索引变量
38. selectedItem **=** item**;**
39. //按上下键是循环显示的，小于0就变成最大值，大于最大值就变成0
40. **if(**selectedItem**<** **0){**
41. selectedItem **=** $autocomplete**.**find**(**"li"**).**length **-** **1;**
42. **}else** **if(**selectedItem**>** $autocomplete**.**find**(**"li"**).**length **-** **1){**
43. selectedItem **=** **0;**
44. **}**
45. //首先移除其他列表项的高亮背景，然后再高亮当前索引背景
46. $autocomplete**.**find**(**"li"**).**removeClass**(**"highlight"**).**eq**(**selectedItem**).**addClass**(**"highlight"**);**
47. **};**
48. varajax\_request **=** **function()** **{**
49. //ajax服务端通信
50. $**.**ajax**(** **{**
51. 'url' **:** 'getResponse.php'**,** //服务器的地址
52. 'data' **:** **{**
53. 'search-text' **:** $searchInput**.**val**()**
54. **},** //参数
55. 'dataType' **:** 'json'**,** //返回数据类型
56. 'type' **:** 'POST'**,** //请求类型
57. 'success' **:** **function(**data**)** **{**
58. **if** **(**data**.**length**)** **{**
59. //遍历data，添加到自动完成区
60. $**.**each**(**data**,** **function(**index**,** term**)** **{**
61. //创建li标签,添加到下拉列表中
62. $**(**'<li></li>'**).**text**(**term**).**appendTo**(**$autocomplete**).**addClass**(**
63. 'clickable'**).**hover**(function()** **{**
64. //下拉列表每一项的事件，鼠标移进去的操作
65. $**(this).**siblings**().**removeClass**(**'highlight'**);**
66. $**(this).**addClass**(**'highlight'**);**
67. selectedItem **=** index**;**
68. **},** **function()** **{**
69. //下拉列表每一项的事件，鼠标离开的操作
70. $**(this).**removeClass**(**'highlight'**);**
71. //当鼠标离开时索引置-1，当作标记
72. selectedItem **=** **-1;**
73. **}).**click**(function()** **{**
74. //鼠标单击下拉列表的这一项的话，就将这一项的值添加到输入框中
75. $searchInput**.**val**(**term**);**
76. //清空并隐藏下拉列表
77. $autocomplete**.**empty**().**hide**();**
78. **});**
79. **});**//事件注册完毕
80. //设置下拉列表的位置，然后显示下拉列表
81. varypos **=** $searchInput**.**position**().**top**;**
82. varxpos **=** $searchInput**.**position**().**left**;**
83. $autocomplete**.**css**(**'width'**,** $searchInput**.**css**(**'width'**));**
84. $autocomplete**.**css**(** **{**
85. 'position' **:** 'relative'**,**
86. 'left' **:** xpos **+** "px"**,**
87. 'top' **:** ypos **+** "px"
88. **});**
89. setSelectedItem**(0);**
90. //显示下拉列表
91. $autocomplete**.**show**();**
92. **}**
93. **}**
94. **});**
95. **};**
96. //对输入框进行事件注册
97. $searchInput
98. **.**keyup**(function(**event**)** **{**
99. //字母数字，退格，空格
100. **if** **(**event**.**keyCode**>** **40** **||** event**.**keyCode **==** **8**
101. **||** event**.**keyCode **==** **32)** **{**
102. //首先删除下拉列表中的信息
103. $autocomplete**.**empty**().**hide**();**
104. clearTimeout**(**timeoutid**);**
105. timeoutid **=** setTimeout**(**ajax\_request**,** **100);**
106. **}** **else** **if** **(**event**.**keyCode **==** **38)** **{**
107. //上
108. //selectedItem = -1 代表鼠标离开
109. **if** **(**selectedItem **==** **-1)** **{**
110. setSelectedItem**(**$autocomplete**.**find**(**'li'**).**length **-** **1);**
111. **}** **else** **{**
112. //索引减1
113. setSelectedItem**(**selectedItem **-** **1);**
114. **}**
115. event**.**preventDefault**();**
116. **}** **else** **if** **(**event**.**keyCode **==** **40)** **{**
117. //下
118. //selectedItem = -1 代表鼠标离开
119. **if** **(**selectedItem **==** **-1)** **{**
120. setSelectedItem**(0);**
121. **}** **else** **{**
122. //索引加1
123. setSelectedItem**(**selectedItem **+** **1);**
124. **}**
125. event**.**preventDefault**();**
126. **}**
127. **}).**keypress**(function(**event**)** **{**
128. //enter键
129. **if** **(**event**.**keyCode **==** **13)** **{**
130. //列表为空或者鼠标离开导致当前没有索引值
131. **if** **(**$autocomplete**.**find**(**'li'**).**length **==** **0**
132. **||** selectedItem **==** **-1)** **{**
133. **return;**
134. **}**
135. $searchInput**.**val**(**$autocomplete**.**find**(**'li'**).**eq**(**
136. selectedItem**).**text**());**
137. $autocomplete**.**empty**().**hide**();**
138. event**.**preventDefault**();**
139. **}**
140. **}).**keydown**(function(**event**)** **{**
141. //esc键
142. **if** **(**event**.**keyCode **==** **27)** **{**
143. $autocomplete**.**empty**().**hide**();**
144. event**.**preventDefault**();**
145. **}**
146. **});**
147. //注册窗口大小改变的事件，重新调整下拉列表的位置
148. $**(**window**).**resize**(function()** **{**
149. varypos **=** $searchInput**.**position**().**top**;**
150. varxpos **=** $searchInput**.**position**().**left**;**
151. $autocomplete**.**css**(**'width'**,** $searchInput**.**css**(**'width'**));**
152. $autocomplete**.**css**(** **{**
153. 'position' **:** 'relative'**,**
154. 'left' **:** xpos **+** "px"**,**
155. 'top' **:** ypos **+** "px"
156. **});**
157. **});**
158. **});**
159. **</**script**>**
160. **</**head**>**
161. **<**body**>**
162. **<**div id**=**"search"**>**
163. **<**label **for=**"search-text"**>**
164. 请输入关键词
165. **</**label**>**
166. **<**input type**=**"text" id**=**"search-text" name**=**"search-text" **/>**
167. **<**input type**=**"button" id**=**"submit" value**=**"搜索" **/>**
168. **</**div**>**
169. **</**body**>**
170. **</**html**>**

getResponse.php完整代码如下：

1. **<?**php
2. $keyword **=** $\_REQUEST**[**"search-text"**];**
3. $words **=** **[**"amani"**,**"abc"**,**"apple"**,**"abstract"**,**"an"**,**
4. "bike"**,**"byebye"**,**"beat"**,**"be"**,**"bing"**,**"come"**,**"cup"**,**"class"**,**"calendar"**,**"chine"**];**
5. $responseWords **=** array**();**
6. foreach**(**$words as $word**){**
7. **if(**strpos**(**$word**,**$keyword**)** **===** **0){**
8. array\_push**(**$responseWords**,**$word**);**
9. **}**
10. **}**
12. echojson\_encode**(**$responseWords**);**
13. **?>**