**1 列举DOM中定义的节点类型和各自的名值**

DOM模型把一篇XML/HTML文档的内容理解为一棵向下生长的树，文档中的每个内容都是这棵树上的一个节点，根据文档内容的不同，可以把这些节点分为哪些类型？每种类型的节点名和节点值各是什么？

**参考答案**

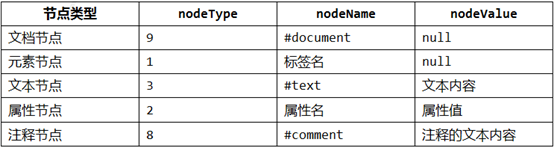


图-1

**2 总结遍历节点树和元素树的相关方法**

**参考答案**

节点树把文档中所有的内容都看做树的一部分，如元素、属性、文本、注释等；

元素树只把文档中的元素看做树的一部分，其它内容加以忽略；故一篇文档的元素树是其对应的节点树的一个子集。

常用方法如图－2所示。

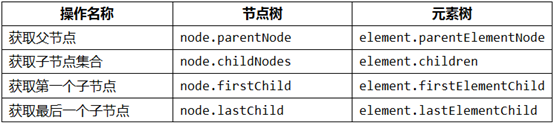


图-2

***注意：元素树相关方法在IE8及以下浏览器中支持性不好。***

**3 全选和取消全选**

CMS系统中，经常需要批量删除数据，这就需要可以进行批量选定记录，点击提交按钮，把所选定的记录id提交给服务器，如图-3：

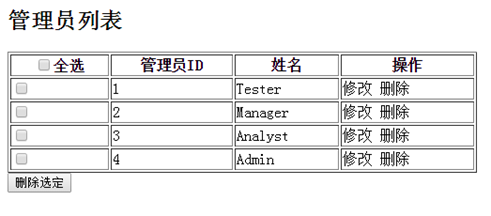


图-3

点击表格中的“全选”复选框，则当前表格中所有的复选框都会被选定；否则所有的复选框都会被取消选择。

**参考答案**

在今天的课程中，选定元素可以有如下三种方法：

(1)getElementById(): 只能根据ID号选择某一个元素

(2)getElementsByTagName():根据标签名选择同类型的元素，但一个表格中可能有多种input而不仅仅是checkbox input，故此方法当前案例中不合适

(3)getElementsByName():根据元素name属性值选择一批元素，由于将选定的记录编号提交给服务器，故需要选定的checkbox的name属性值必然相同，此方法此情景下最适用。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise01\_01.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**全选和取消全选**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript文件引用，创建静态数据表格**

除了“全选”复选框外，表格中其它复选框都需要name属性，且name属性值一样。“全选”复选框需要指定单击事件处理方法。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**全选和取消全选**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise01\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<h2>管理员列表</h2>**
10. **<table border=**"1px" **width=**"500px"**>**
11. **<thead>**
12. **<tr>**
13. **<th><input type=**"checkbox" **onclick=**"selectAll(this)"/>全选</**th>**
14. **<th>管理员ID</th>**
15. **<th>姓名</th>**
16. **<th>操作</th>**
17. **</tr>**
18. **</thead>**
19. **<tbody>**
20. **<tr>**
21. **<td><input type=**"checkbox" **name=**"adminID"/></**td>**
22. **<td>1</td>**
23. **<td>Tester</td>**
24. **<td>修改 删除</td>**
25. **</tr>**
26. **<tr>**
27. **<td><input type=**"checkbox" **name=**"adminID"/></**td>**
28. **<td>2</td>**
29. **<td>Manager</td>**
30. **<td>修改 删除</td>**
31. **</tr>**
32. **<tr>**
33. **<td><input type=**"checkbox" **name=**"adminID"/></**td>**
34. **<td>3</td>**
35. **<td>Analyst</td>**
36. **<td>修改 删除</td>**
37. **</tr>**
38. **<tr>**
39. **<td><input type=**"checkbox" **name=**"adminID"/></**td>**
40. **<td>4</td>**
41. **<td>Admin</td>**
42. **<td>修改 删除</td>**
43. **</tr>**
44. **</tbody>**
45. **</table>**
46. **<button>删除选定</button>**
47. **</**body**>**
48. **</**html**>**

**步骤三：创建外部JavaScript文件exercise01\_01.js，创建selectAll()函数**

1. //设置全选或取消全选
2. //需要参数cb：即“全选”复选框，函数需要知道此复选框是否处于选定状态
3. **function** selectAll**(**cb**){**
4. //选择数据表中的所有记录行前的checkbox
5. **var** cbList **=** document**.**getElementsByName**(**'adminID'**);**
7. **for(var** i**=0;** i**<**cbList**.**length**;** i**++){**
8. cbList**[**i**].**checked **=** cb**.**checked**;**
9. **}**
10. **}**

**4 隐藏选定DIV**

页面中有若干个显示数据的区块，如图-4所示：

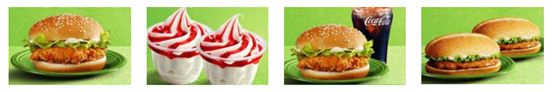


图-4

单击某区块，可以选定该元素（以红色边框突出显示）；再次单击，可以取消选定，如图-5所示：

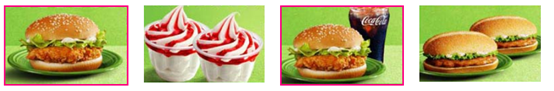


图-5

单击“隐藏选定”按钮，可以将选定的元素隐藏起来，如图-6所示：



图-6

忽略IE8及以下版本浏览器的兼容性问题，实现上述需求。

**参考答案**

由于本案例需求中允许忽略IE8以下浏览器的兼容性问题，就可以使用类似CSS中的选择器来选定要操作的元素；选定的元素需要使用红色边框突出显示，可以通过修改元素所属的class来实现，接下来再选定这些具有特定class的元素，修改样式即可。故本案例使用getElementsByClassName()以及querySelector()、querySelectorAll()方法实现起来最方便。

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 exercise01\_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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**隐藏选定DIV**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript和CSS文件引用，**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**隐藏选定DIV**</**title**>**
6. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise01\_02.css" **/>**
7. **<script type=**"text/javascript" **src=**"exercise01\_02.js"**></script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：添加HTML元素**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**隐藏选定DIV**</**title**>**
6. **<**link rel**=**"stylesheet" type**=**"text/css" href**=**"exercise01\_02.css" **/>**
7. **<**script type**=**"text/javascript" src**=**"exercise01\_02.js"**></**script**>**
8. **</**head**>**
9. **<**body**>**
10. **<h2>隐藏选定DIV</h2>**
11. **<button>隐藏选定</button>**
12. **<div id=**"container"**>**
13. **<div><img src=**"img/1.png"/></**div>**
14. **<div><img src=**"img/2.png"/></**div>**
15. **<div><img src=**"img/3.png"/></**div>**
16. **<div><img src=**"img/4.png"/></**div>**
17. **</div>**
18. **</**body**>**
19. **</**html**>**

**步骤四：创建外部CSS样式文件exercise01\_02.css**

1. /\*每个展示数据的div的默认样式\*/
2. #container div**{**
3. background**-**color**:** #80ff80**;**
4. margin**:** 10px**;**
5. float**:**left**;**
6. **}**
7. /\*当展示数据的div处于被选定状态时的样式\*/
8. **.**chosen**{**
9. border**:** 2px solid #ff0080**;**
10. **}**

**步骤五：创建外部JavaScript文件exercise01\_02.js，声明bind()函数**

1. //当页面加载完成时，为按钮和div绑定单击事件处理方法
2. **function** bind**(){**
3. //为按钮绑定事件处理方法
4. **var** bt **=** document**.**querySelector**(**'button'**);**    //根据标签名选取第一个元素
5. bt**.**onclick **=** hideChosen**;**
7. //为可以选定的div绑定单击事件处理方法
8. **var** divList **=** document**.**querySelectorAll**(**'#container div'**);**    //根据复合选择器选取多个元素
9. **for(var** i**=0;** i**<**divList**.**length**;** i**++){**
10. divList**[**i**].**onclick **=** toggleChosen**;**
11. **}**
12. **}**

上述代码中，为“隐藏选定”的单击事件绑定一个hideChosen()函数；为每个呈现数据的div的单击事件绑定了一个toggleChosen()方法，下面继续编写这两个方法：

1. //当div被单击时，执行选定/取消选定设置
2. **function** toggleChosen**(){**
3. //this此时指向被单击的div元素
4. **if(this.**getAttribute**(**'class'**)==**'chosen'**){**
5. **this.**removeAttribute**(**'class'**);**
6. **}else{**
7. **this.**setAttribute**(**'class'**,** 'chosen'**);**
8. **}**
9. **}**
10. //隐藏选定div
11. **function** hideChosen**(){**
12. **var** chosenDivs **=** document**.**getElementsByClassName**(**'chosen'**);**
13. **if(**chosenDivs**.**length**==0){**
14. alert**(**'请选择至少一个DIV'**);**
15. **}else{**
16. **for(var** i**=0;** i**<**chosenDivs**.**length**;** i**++){**
17. console**.**log**(**chosenDivs**[**i**]);**
18. chosenDivs**[**i**].**style**.**display **=** 'none'**;**
19. **}**
20. **}**
21. **}**

exercise01\_02.js文件中完整内容如下：

1. //当页面加载完成时，为按钮和div绑定单击事件处理方法
2. **function** bind**(){**
3. //为按钮绑定事件处理方法
4. **var** bt **=** document**.**querySelector**(**'button'**);**    //根据标签名选取第一个元素
5. bt**.**onclick **=** hideChosen**;**
7. //为可以选定的div绑定单击事件处理方法
8. **var** divList **=** document**.**querySelectorAll**(**'#container div'**);**    //根据复合选择器选取多个元素
9. **for(var** i**=0;** i**<**divList**.**length**;** i**++){**
10. divList**[**i**].**onclick **=** toggleChosen**;**
11. **}**
12. **}**
13. //当div被单击时，执行选定/取消选定设置
14. **function** toggleChosen**(){**
15. //this此时指向被单击的div元素
16. **if(this.**getAttribute**(**'class'**)==**'chosen'**){**
17. **this.**removeAttribute**(**'class'**);**
18. **}else{**
19. **this.**setAttribute**(**'class'**,** 'chosen'**);**
20. **}**
21. **}**
22. //隐藏选定div
23. **function** hideChosen**(){**
24. **var** chosenDivs **=** document**.**getElementsByClassName**(**'chosen'**);**
25. **if(**chosenDivs**.**length**==0){**
26. alert**(**'请选择至少一个DIV'**);**
27. **}else{**
28. **for(var** i**=0;** i**<**chosenDivs**.**length**;** i**++){**
29. console**.**log**(**chosenDivs**[**i**]);**
30. chosenDivs**[**i**].**style**.**display **=** 'none'**;**
31. **}**
32. **}**
33. **}**

**1 简述核心DOM与HTML DOM的关系**

**参考答案**

DOM是W3C组织制定的行业标准，分为三个部分：

(1)核心 DOM - 针对任何结构化文档的标准模型；

(2)XML DOM - 针对 XML 文档的标准模型；

(3)HTML DOM - 针对 HTML 文档的标准模型。

其中，核心DOM定义了操作树形结构化文档的标准方法，如创建、查找、删除、遍历节点。HTML DOM是核心DOM的扩展，其提供的方法尤其适合于读取或修改元素的属性。

**2 总结读取一个p标签中的文本内容有哪些方法**

创建一个HTML页面，演示可以使用哪些方法读取一个p标签中的文本内容。注意提示出浏览器兼容性问题。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_01.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**总结读取一个p标签中的文本内容有哪些方法**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加一个待查询的目标<p>元素**

为<p>元素增加id属性，稍后可以使用document.getElementById()方法找到此元素；同时，为了更具代表性，子节点中特意添加了一个<b>元素

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**总结读取一个p标签中的文本内容有哪些方法**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **<p id=**"content"**>前置内容<b>强调内容<**/b>后置内容</**p>**
9. **</**body**>**
10. **</**html**>**

**步骤三：创建script块，直接在页面中进行测试**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**总结读取一个p标签中的文本内容有哪些方法**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **<**p id**=**"content"**>**前置内容**<**b**>**强调内容**<**/b>后置内容</p**>**
10. **<**script type**=**"text/javascript"**>**
11. **var** p **=** document**.**getElementById**(**'content'**);**
13. //方式一：使用元素的innerHTML属性
14. console**.**log**(** p**.**innerHTML **);**
15. //输出“前置内容<b>强调内容</b>后置内容”
17. //方式二：使用textContent属性
18. console**.**log**(** p**.**textContent **);**
19. //输出“前置内容强调内容后置内容”，IE8及以下版本不支持
21. //方式三：使用innerText属性
22. console**.**log**(** p**.**innerText **);**
23. //输出“前置内容强调内容后置内容”，Firefox不支持
25. //方式四：使用父子关系
26. **var** txt **=** ''**;**
27. **for(var** i**=0;** i**<**p**.**childNodes**.**length**;** i**++){**
28. txt **+=** p**.**childNodes**[**i**].**nodeValue**;**
29. **}**
30. console**.**log**(** txt **);**
31. //输出“前置内容null后置内容”，浏览器兼容性好，但对于子元素需要单独的处理，甚至递归的处理
32. **</**script**>**
33. **</**body**>**
34. **</**html**>**

**3 表格版日历**

制作表格版日历，页面初始化时，效果如图－1所示。

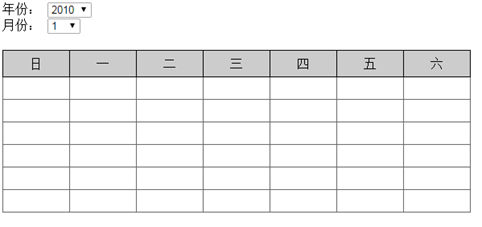
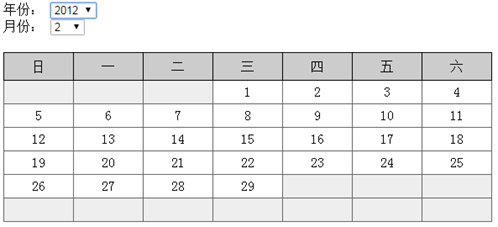


图-1

选择某年份以及某月份后，则显示日历，效果如图-2和图－3所示。



图－2

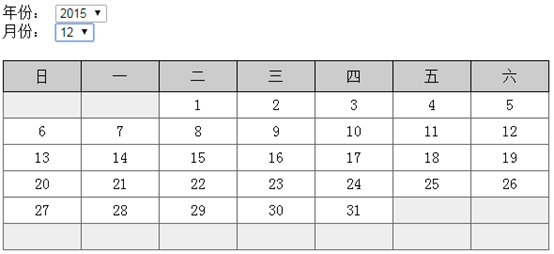


图-3

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**

**步骤二：为页面添加外部JavaScript文件和CSS文件的引用**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise02\_02.css" **/>**
7. **<script type=**"text/javascript" **src=**"exercise02\_02.js"**></script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**

**步骤三：创建页面内容**

创建如图－1所示的页面内容，代码如下所示。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**日历**</**title**>**
5. **<**meta charset**=**"utf-8" **/>**
6. **<**link rel**=**"Stylesheet" href**=**"exercise02\_02.css" **/>**
7. **<**script type**=**"text/javascript" src**=**"exercise02\_02.js"**></**script**>**
8. **</**head**>**
9. **<**body**>**
10. **年份：**
11. **<select id=**"year" **onchange=**"showDate();"**>**
12. **<option>2010</option>**
13. **<option>2011</option>**
14. **<option>2012</option>**
15. **<option>2013</option>**
16. **<option>2014</option>**
17. **<option>2015</option>**
18. **<option>2016</option>**
19. **<option>2017</option>**
20. **</select>**
21. **<br />**
22. **月份：**
23. **<select id=**"month" **onchange=**"showDate();"**>**
24. **<option>1</option>**
25. **<option>2</option>**
26. **<option>3</option>**
27. **<option>4</option>**
28. **<option>5</option>**
29. **<option>6</option>**
30. **<option>7</option>**
31. **<option>8</option>**
32. **<option>9</option>**
33. **<option>10</option>**
34. **<option>11</option>**
35. **<option>12</option>**
36. **</select>**
37. **<table id=**"date"**>**
38. **<thead>**
39. **<tr>**
40. **<td>日<**/td><td>一</**td><td>二<**/td><td>三</**td><td>四<**/td><td>五</**td><td>六</td>**
41. **</tr>**
42. **</thead>**
43. **<tbody>**
44. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
45. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
46. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
47. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
48. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
49. **<tr><td><**/td><td></**td><td><**/td><td></**td><td><**/td><td></**td><td><**/td></**tr>**
50. **</tbody>**
51. **</table>**
52. **</**body**>**
53. **</**html**>**

**步骤四：创建外部CSS文件，为表格定义样式**

打开文件exercise02\_02.css，为表格定义样式。代码如下：

1. #date **{**
2. border**-**collapse**:**collapse**;**
3. margin**-**top**:**20px**;**
4. **}**
5. #date thead td **{**
6. border**:**1px solid #**000;**
7. background**-**color**:**#ccc**;**
8. width**:**80px**;**
9. height**:**30px**;**
10. text**-**align**:**center**;**
11. **}**
12. #date tbody td **{**
13. border**:**1px solid #**666;**
14. height**:**25px**;**
15. text**-**align**:**center**;**
16. **}**

**步骤五：创建外部JavaScript文件，定义函数计算天数**

打开文件exercise02\_02.js，定义函数，计算某年某月的天数。需要考虑闰年以及大月和小月的问题。代码如下：

1. //根据年份和月份，得到该月的天数
2. **function** getDays**(**year**,** month**)** **{**
3. **if** **(**month **==** **4** **||** month **==** **6** **||** month **==** **9** **||** month **==** **11)**
4. **return** **30;**
5. **else** **if** **(**month **==** **2)** **{**
6. **if((**year **%** **4** **==** **0** **&&** year **%** **100** **!=** **0)** **||** year **%** **400==0** **)**
7. **return** **29;**
8. **else**
9. **return** **28;**
10. **}**
11. **else**
12. **return** **31;**
13. **}**

**步骤六：定义函数，在表格中显示天数**

打开文件exercise02\_02.js，继续定义函数，根据天数在页面上显示日期。代码如下：

1. **function** showDate**()** **{**
2. //得到年份
3. **var** yearNode **=** document**.**getElementById**(**"year"**);**
4. **var** yearIndex **=** yearNode**.**selectedIndex**;**
5. **var** year **=** yearNode**.**getElementsByTagName**(**"option"**)[**yearIndex**].**innerHTML**;**
6. //得到月份
7. **var** monthNode **=** document**.**getElementById**(**"month"**);**
8. **var** monthIndex **=** monthNode**.**selectedIndex**;**
9. **var** month **=** monthNode**.**getElementsByTagName**(**"option"**)[**monthIndex**].**innerHTML**;**
10. //某年某月的天数
11. **var** days **=** getDays**(**year**,** month**);**
12. **var** dateObj **=** **new** Date**(**year**,** month**-1,** **1);**
13. //最多６行 7 列，创建并初始化一个 42 长度的数组
14. **var** arr **=** **[];**
15. **var** date **=** **1;**
16. **for** **(var** i **=** **0;** i **<** **42;** i**++)** **{**
17. //１号之前以及最大日期之后，都记载为 0；其他记载为日期
18. **if** **(**i **<** dateObj**.**getDay**()** **||** date **>** days**)**
19. arr**[**i**]** **=** **0;**
20. **else** **{**
21. arr**[**i**]** **=** date**;**
22. date**++;**
23. **}**
24. **}**
25. //根据数组初始化页面表格 ：表头行除外，６行７列
26. **var** cells **=** document**.**getElementById**(**"date"**).**getElementsByTagName**(**"tbody"**)[0].**getElementsByTagName**(**"td"**);**
27. //得到 42 个单元格，根据数组写入日期
28. **for** **(var** i **=** **0;** i **<** cells**.**length**;** i**++)** **{**
29. **if** **(**arr**[**i**]** **!=** **0)** **{**
30. cells**[**i**].**innerHTML **=** arr**[**i**];**
31. cells**[**i**].**style**.**backgroundColor **=** "#fff"**;**
32. **}**
33. **else** **{**
34. cells**[**i**].**innerHTML **=** ""**;**
35. cells**[**i**].**style**.**backgroundColor **=** "#eee"**;**
36. **}**
37. **}**
38. **}**

**1 批量提交数据**

创建页面，效果如图－1所示：



图－1

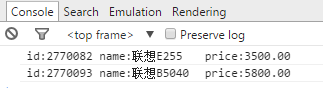
单击页面上的“增加产品“按钮，则在表格中增加一行，行中的各单元格显示文本框，用于录入相应的数据，界面效果如图－2所示：



图－2

录入完毕后，单击页面上的”保存“按钮，则使用自定义对象（三种方式任选一种）封装每个产品数据，最后形成对象的数组以备提交。其中，每个产品对象需要有id属性、name属性和 price 属性。

另外，此案例中只能模拟提交的效果，因此，在将数据封装完毕并记载成为数组之后，需要循环对象的数组，逐一读取对象中的数据，在控制台输出进行测试。效果如图－3：



图－3

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_01.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**批量提交数据**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部 JavaScript和 CSS 文件引用**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**批量提交数据**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise03\_01.css" **/>**
7. **<script language=**"javascript" **src=**"exercise03\_01.js" **type=**"text/javascript"**></script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：添加页面基本元素**

***注意：为了避免文本子元素的干扰，tbody的开始和结束标记紧挨着写在一行中。***

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**批量提交数据**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**link rel**=**"stylesheet" type**=**"text/css" href**=**"exercise03\_01.css" **/>**
7. **<**script language**=**"javascript" src**=**"exercise03\_01.js" type**=**"text/javascript"**></**script**>**
8. **</**head**>**
9. **<**body**>**
10. **<h2>批量提交数据</h2>**
11. **<table id=**"dataTable"**>**
12. **<thead>**
13. **<tr>**
14. **<th>产品 ID</td>**
15. **<th>产品名称</td>**
16. **<th>产品价格</td>**
17. **</tr>**
18. **</thead>**
19. **<tbody></tbody>**
20. **</table>**
21. **<br />**
22. **<input type=**"button" **value=**"增加产品" **onclick=**"addNewRow()" **/>**
23. **<input type=**"button" **value=**"保存" **onclick=**"saveData()" **/>**
24. **</**body**>**
25. **</**html**>**

**步骤四：创建外部CSS文件exercise03\_01.css**

1. /\*表格\*/
2. table**{**
3. border**-**collapse**:**collapse**;**
4. **}**
5. td**,**th**{**
6. border**:**1px solid gray**;**
7. width**:**150px**;**
8. height**:**30px**;**
9. text**-**align**:**center**;**
10. **}**
11. th**{**
12. height**:**35px**;**
13. font**-**weight**:**bold**;**
14. background**-**color**:**#ddd**;**
15. **}**
16. td input**{**
17. width**:**120px**;**
18. **}**

**步骤五：创建外部JavaScript文件exercise03\_01.js**

1. //用于封装产品数据的自定义对象
2. **function** Product**(**id**,** n**,** p**)** **{**
3. // 属性
4. **this.**id **=** id**;**
5. **this.**name **=** n**;**
6. **this.**price **=** p**;**
7. // 方法
8. **this.**show **=** **function()** **{**
9. console**.**log**(**'id:' **+** **this.**id **+** '**\t**name:' **+** **this.**name **+** '**\t**price:' **+** **this.**price**);**
10. **};**
11. **}**
12. //为表格添加行
13. **function** addNewRow**()** **{**
14. // 为行创建 id 单元格
15. **var** idCell **=** document**.**createElement**(**'td'**);**
16. **var** input1 **=** document**.**createElement**(**'input'**);**
17. idCell**.**appendChild**(**input1**);**
18. // 为行创建 name 单元格
19. **var** nameCell **=** document**.**createElement**(**'td'**);**
20. **var** input2 **=** document**.**createElement**(**'input'**);**
21. nameCell**.**appendChild**(**input2**);**
22. // 为行创建 price 单元格
23. **var** priceCell **=** document**.**createElement**(**'td'**);**
24. **var** input3 **=** document**.**createElement**(**'input'**);**
25. priceCell**.**appendChild**(**input3**);**
26. // 创建新行
27. **var** row **=** document**.**createElement**(**'tr'**);**
28. row**.**appendChild**(**idCell**);**
29. row**.**appendChild**(**nameCell**);**
30. row**.**appendChild**(**priceCell**);**
31. // 查找表格及表体对象
32. **var** table **=** document**.**getElementById**(**'dataTable'**);**
33. **var** tbody **=** table**.**getElementsByTagName**(**'tbody'**)[0];**
34. tbody**.**appendChild**(**row**);**
35. **}**
36. // 收集表格中的数据
37. **function** saveData**()** **{**
38. // 定义数组，用于存储数据
39. **var** datas **=** **[];**
40. // 循环表格，并收集数据
41. **var** table **=** document**.**getElementById**(**'dataTable'**);**
42. **var** tbody **=** table**.**getElementsByTagName**(**'tbody'**)[0];**
43. **var** rows **=** tbody**.**childNodes**;**
44. **for** **(var** i **=** **0;** i **<** rows**.**length**;** i**++)** **{**
45. // 获得页面的数据
46. **var** id **=** rows**[**i**].**firstChild**.**firstChild**.**value**;**
47. **var** name **=** rows**[**i**].**firstChild**.**nextSibling**.**firstChild**.**value**;**
48. **var** price **=** rows**[**i**].**lastChild**.**firstChild**.**value**;**
49. // 定义对象以封装数据
50. **var** p **=** **new** Product**(**id**,** name**,** price**);**
51. datas**[**i**]** **=** p**;**
52. **}**
53. // 提交数据（测试数据）
54. **for** **(var** i **=** **0;** i **<** datas**.**length**;** i**++)** **{**
55. datas**[**i**].**show**();**
56. **}**
57. **}**

**2 举例说明文档片段的作用**

**参考答案**

通过使用var fragment = document.createDocumentFragement()可以创建一个文档片段对象。

该对象在页面中没有任何显示内容或对应的HTML标签，它的存在仅仅为了包含新创建的其它DOM对象。

假设现在需要使用js连续为一个table添加10个tr元素，如果执行类似如下的循环语句：

1. **for(var** i**=0;** i**<10;** i**++){**
2. table**.**appendChild**(**tr**);**
3. **}**

那么，每当页面中添加一个新元素，浏览器都要重新绘制一遍完整的DOM树，由于这个过程需要一段时间（如0.x秒），在这段时间内，浏览器的一部分就显示为空白，这样循环下来，在连续添加10个tr的过程中，整个页面好像频闪了10次。

若使用了文档片段对象，则上述代码可以重写为：

1. **for(var** i**=0;** i**<10;** i**++){**
2. fragment**.**appendChild**(**tr**);**
3. **}**
4. table**.**appendChild**(**fragment**);**

循环过程中，10个tr对象只是在内存中追加为fragment的子节点，并未追加到table中，故也就不会被绘制到窗口中；待循环执行完毕，再一次性的文档片段追加到table中，这样只需要一次绘制的过程，所有的tr就都被追加到页面中了，提高了绘制的效率，减少了页面的频闪(只有一次了)。

**3 创建动态表格**

创建一个具有广泛通用性的表格数据加载方法，表格列名、列的数量、记录行的数量都可以实际数据量而改变。且记录行实现隔行变色，鼠标悬停时深色着重显示。如图-4所示：

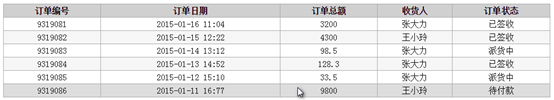


图-4

**参考答案**

由于表中的数据量无法预估，那么久必须对获取的数据有一定的格式方面的要求；例如，可以自定义待加载的数据必须满足如下格式：

1. **[**
2. "colNames" **:** **[**'xx'**,** **...**'zz'**],**
3. "rows" **:** **[**
4. **[**"a"**,...**"aaa"**],**
5. **...**
6. **[**"c"**,...**"ccc"**]**
7. **]**
8. **]**

这种特定的数据格式可以与数据提供者（如服务器端数据提供页面）进行协调，若实在无法提供类似上述格式，则需要对数据进行格式转换了（数据适配）。本案例假设页面得到的数据刚好满足上述格式要求。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_03.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**创建动态表格**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript和CSS文件引用，添加页面元素**

原始表格中创建一个简单的提示消息“暂无数据”，需要点击“加载数据”按钮才可以加载并解析数据，故需要为按钮绑定单击事件处理方法。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**创建动态表格**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise03\_03.css"**/>**
7. **<script type=**"text/javascript"**src=**"exercise03\_03.js"**></script>**
8. **</**head**>**
9. **<**body**>**
10. **<h2>创建动态表格</h2>**
11. **<button onclick=**"loadData(data)"**>加载数据<**/button><br/**><br/>**
12. **<table id=**"dataTable"**><tbody><**/tbody><tr><td>暂无数据</**td><**/tr></**table><br/>**
13. **提示：<br/>**
14. **(1)**点击“加载数据”按钮可以加载任何符合一定格式要求的数据**<**br**/>**
15. **(2)**待加载数据中列名、列的数量、记录行的都是都是可变的**<**br**/>**
16. **(3)**显示出来的数据隔行变色
17. **(4)**记录行鼠标悬停时变色显示
18. **</**body**>**
19. **</**html**>**

**步骤三：创建外部CSS样式文件exercise03\_03.css**

1. #dataTable**{**
2. border**:** 1px solid #aaaaaa**;**
3. width**:** **80%;**
4. border**-**collapse**:** collapse**;**
5. **}**
6. th**,**td**{**
7. border**:** 1px solid #aaaaaa**;**
8. text**-**align**:** center**;**
9. padding**:** 3px**;**
10. **}**
11. thead tr**{**
12. background**-**color**:** #d0d0d0**;**
13. **}**
14. **.**even**{**
15. background**-**color**:** #f6f6f6**;**
16. **}**

**步骤四：创建外部JavaScript文件exercise03\_03.js**

首先创建一个符合预期格式的数据对象，其实这个数据对象完全可以不是静态的，而是临时从服务器端某个URL加载得到的（AJAX技术）；接下来创建loadData()函数处理这些数据。

1. //模拟从服务器获取到的具有特定格式的JSON数据
2. **var** data **=** **{**
3. "colNames"**:[**'订单编号'**,**'订单日期'**,**'订单总额'**,**'收货人'**,** '订单状态'**],**
4. "rows"**:[**
5. **[9319081,** '2015-01-16 11:04'**,** **3200,** '张大力'**,** '已签收'**],**
6. **[9319082,** '2015-01-15 12:22'**,** **4300,** '王小玲'**,** '已签收'**],**
7. **[9319083,** '2015-01-14 13:12'**,** **98.5,** '张大力'**,** '派货中'**],**
8. **[9319084,** '2015-01-13 14:52'**,** **128.3,** '张大力'**,** '已签收'**],**
9. **[9319085,** '2015-01-12 15:10'**,** **33.5,** '张大力'**,** '派货中'**],**
10. **[9319086,** '2015-01-11 16:77'**,** **9800,** '王小玲'**,** '待付款'**]**
11. **]**
12. **};**
13. //可以加载任意规整格式数据的表格
14. **function** loadData**(**data**){**
15. **if(**data**){**
16. **var** table **=** document**.**getElementById**(**'dataTable'**);**
17. clear**(**table**)**    //只要实参中有数据，则删除表中原有内容
18. createTHead**(**table**,** data**.**colNames**);**    //创建第一行，即表头
19. createTBody**(**table**,** data**.**rows**);**        //创建表中的记录行
20. **}**
21. **}**
22. **function** clear**(**table**){**
23. **while(**table**.**firstChild**){**
24. table**.**removeChild**(**table**.**firstChild**);**
25. **}**
26. **}**
27. **function** createTHead**(**table**,** cols**){**
28. //若有thead，则返回；否则创建新的thead并添加到table
29. **var** thead **=** table**.**createTHead**();**
30. **var** tr **=** thead**.**insertRow**();**
31. **for(var** i**=0;** i**<**cols**.**length**;** i**++){**
32. **var** td **=** document**.**createElement**(**'th'**);**//tr.insertRow()只能创建td无法创建th
33. td**.**innerHTML **=** cols**[**i**];**
34. tr**.**appendChild**(**td**);**
35. **}**
36. **}**
37. **function** createTBody**(**table**,** rows**){**
38. **var** tbody **=** document**.**createElement**(**'tbody'**);**//table.createTBody()IE不支持
39. table**.**appendChild**(**tbody**);**
40. **for(var** i**=0;** i**<**rows**.**length**;** i**++){**
41. //在tbody中添加一个tr
42. **var** tr **=** tbody**.**insertRow**();**
43. //隔行变色
44. **if(**i**%2==1){**
45. tr**.**className **=** 'even'**;**
46. **}**
47. **var** oldColor **=** **null;**
48. //鼠标悬停着重强调
49. tr**.**onmouseover **=** **function(){**
50. oldColor **=** **this.**style**.**backgroundColor**;**    //保存原始颜色
51. **this.**style**.**backgroundColor **=** '#dddddd'**;**
52. **};**
53. //鼠标离开恢复原色
54. tr**.**onmouseout **=** **function(){**
55. **this.**style**.**backgroundColor **=** oldColor**;**    //恢复原始颜色
56. **}**
57. //向tr中添加td
58. **for(var** j**=0;** j**<**rows**[**i**].**length**;** j**++){**
59. **var** td **=** tr**.**insertCell**();**
60. td**.**innerHTML **=** rows**[**i**][**j**];**
61. **}**
62. **}**
63. **}**

**4 创建可排序可拖拽的动态表格（提高题，选做）**

在今天的课后练习第2题的基本需求之上，继续实现下述两个功能点：

(1)点击表头中的某一列，则表中所有的记录行可以按照此列来进行升序排列；若再点击一次，则所有记录行按照此列的降序来排列。如图-5和图-6所示。

(2)对某一行中的第一列进行拖拽，可以与上面的记录行或下面的记录行进行交换。如图-7所示。

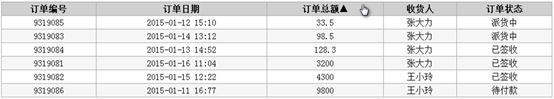


图-5

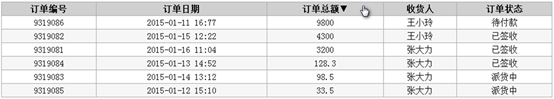


图-6

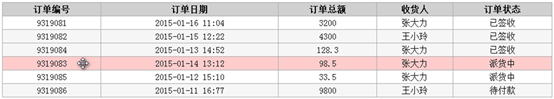


图-7

**参考答案**

“单击表头单元格实现依此排序”功能中，记录用户点击事件所在列的下标，在原始数据中根据此下标列上的值重新排序全部数据，然后再重新绘制表格内容即可——由于原始数据此后仍然需要保留，则最好对原始数据对象进行深度克隆，仅对克隆对象进行重新排序排序即可。此外，由于每行记录都可能包含多项内容，故对记录行进行排序时，需要使用arrayObject.sort( function )方法，其中的参数function用于指定排序的依据。

“拖拽记录行实现重新排序”功能中，需要在鼠标按下时，记录下当前行即为要移动的行；当鼠标移动到其它记录行上方时，在Table对象上查找此行与要移动行的先后顺序，使用table.insertBefore(tr)即可实现两行记录显示效果上的重新排序。注意，此时的重新排序其实并未对表格中的数据本身产生任何影响，仅仅是在DOM树的显示上发生了改变而已。

实现此案例可以按如下步骤进行

**步骤一：修改exercise03\_04.html文件，增加功能提示内容**

将exercise03\_04.html、exercise03\_03.js、exercise03\_03.css分别复制为exercise03\_04.html、exercise03\_04.js、exercise03\_04.css。修改exercise03\_04.html，增加如下代码：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**创建动态表格**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**link rel**=**"stylesheet" type**=**"text/css" href**=**"exercise03\_04.css"**/>**
7. **<**script type**=**"text/javascript"src**=**"exercise03\_04.js"**></**script**>**
8. **</**head**>**
9. **<**body**>**
10. **<**h2**>**创建动态表格**</**h2**>**
11. **<**button onclick**=**"loadData(data)"**>**加载数据**<**/button><br/**><**br**/>**
12. **<**table id**=**"dataTable"**><**tr**><**td**>**暂无数据**<**/td></tr**><**/table><br/**>**
13. 提示：**<**br**/>**
14. **(1)**点击“加载数据”按钮可以加载任何符合一定格式要求的数据**<**br**/>**
15. **(2)**待加载数据中列名、列的数量、记录行的都是都是可变的**<**br**/>**
16. **(3)**显示出来的数据隔行变色**<**br **/>**
17. **(4)**记录行鼠标悬停时变色显示**<**br **/>**
18. **(5)**单击某列的列名时，表中的所有数据可以按照此列上的值升序排列；再次单击此列名时，所有数据又可以按此列上的值降序排列**<**br**/>**
19. **(6)**拖拽某行记录的第一个单元格，可以将其移动到上一行或者下一行
20. **</**body**>**
21. **</**html**>**

**步骤二：修改exercise03\_04.js文件，为表头中每个列的列名部分添加单击事件处理方法**

修改createTable()函数为如下内容：

1. // 创建表头部
2. **function** createTHead**(**table**,** data**)** **{**
3. // 若有thead，则返回；否则创建新的thead并添加到table
4. **var** thead **=** table**.**createTHead**();**
5. **var** tr **=** thead**.**insertRow**();**
6. **for** **(var** i **=** **0;** i **<** data**.**colNames**.**length**;** i**++)** **{**
7. **var** td **=** document**.**createElement**(**'th'**);**// tr.insertRow()只能创建td无法创建th
8. td**.**innerHTML **=** data**.**colNames**[**i**];**
9. **td.id =** 'col' **+ i;** // 为每列添加一个id属性
10. tr**.**appendChild**(**td**);**
11. **}**
12. //若data中指定了需要进行排序的列序号及排序方式
13. **if(data.asc){**
14. **var sortedTd = tr.childNodes[data.sortedBy];**
15. **sortedTd.innerHTML = sortedTd.innerHTML+(data.asc==1?**'▲'**:**'▼'**);**
16. **}**
17. // 为所有的tr增加“点击排序”功能
18. // 注意此处的技巧：没有单独为每个td添加单击处理函数，而是为整个tr添加了单击事件处理函数
19. **tr.style.cursor =** 'pointer'**;**
20. **tr.onclick = function(event) {**
21. **event = event || window.event;**
22. **var td = event.srcElement || event.target;**
23. **var index = td.id.**substring**(3);** //td的id值形如“col3”，index即为被点选的列在原始数据中的顺序
24. **var sortedData =** cloneObject**(data);**//将data进行深度复制，得到完全相同的一个副本
25. **sortedData.sortedBy = index;**//接下来的排序要以哪一列为标准
26. **if(td.innerHTML.**charAt**(td.innerHTML.length-1)==**'▲'**){**//若指定的列名以▲结尾
27. **sortedData.asc = -1;**//-1表降序排列
28. **}else{**
29. **sortedData.asc = 1;**//1表升序排列
30. **}**
31. **sortedData.rows.**sort**(function(row1, row2) {**
32. **if( row1[index] > row2[index]){**
33. **return 1\*sortedData.asc;**
34. **}else if( row1[index] < row2[index]){**
35. **return -1\*sortedData.asc;**
36. **}else {**
37. **return 0;**
38. **}**
39. **});**
40. loadData**(sortedData);**
41. **}**
42. **}**

上述代码中，为了实现将原有数据进行深度克隆的功能，需要创建cloneObject()函数，如下所示：

1. //对象的深度复制
2. **function** cloneObject**(**obj**)** **{**
3. **var** o **=** obj**.**constructor **===** Array **?** **[]** **:** **{};**
4. **for** **(** **var** i **in** obj**)** **{**
5. **if** **(**obj**.**hasOwnProperty**(**i**))** **{**
6. o**[**i**]** **=** **typeof** obj**[**i**]** **===** "object" **?** cloneObject**(**obj**[**i**])** **:** obj**[**i**];**
7. **}**
8. **}**
9. **return** o**;**
10. **}**

**步骤三：继续修改exercise03\_04.js，为表体中每行的第一个td添加拖拽操作相关的监听方法**

修改createTBody()函数，添加如下着重强调的内容：

1. // 创建表体中的数据
2. **function** createTBody**(**table**,** rows**)** **{**
3. **var** tbody **=** document**.**createElement**(**'tbody'**);**// table.createTBody()IE不支持
4. table**.**appendChild**(**tbody**);**
5. **for** **(var** i **=** **0;** i **<** rows**.**length**;** i**++)** **{**
6. // 在tbody中添加一个tr
7. **var** tr **=** tbody**.**insertRow**();**
8. // 隔行变色
9. **if** **(**i **%** **2** **==** **1)** **{**
10. tr**.**className **=** 'even'**;**
11. **}**
12. **var** oldColor **=** **null;**
13. // 鼠标悬停着重强调
14. tr**.**onmouseover **=** **function()** **{**
15. oldColor **=** **this.**style**.**backgroundColor**;** // 保存原始颜色
16. **this.**style**.**backgroundColor **=** '#dddddd'**;**
17. **};**
18. // 鼠标离开恢复原色
19. tr**.**onmouseout **=** **function()** **{**
20. **this.**style**.**backgroundColor **=** oldColor**;** // 恢复原始颜色
21. **}**
22. //将要进行移动的tr
23. **var trToBeMoved = null;**
24. // 向tr中添加td
25. **for (var j = 0; j < rows[i].length; j++) {**
26. **var td = tr.**insertCell**();**
27. **td.innerHTML = rows[i][j];**
28. **if(j==0){**//当用户拖拽第0列时，可以对记录行进行重新排序
29. **td.style.cursor =** 'pointer'**;**
30. **td.onmousedown = function(){**//当鼠标按下时，记录当前行即为要进行移动的行
31. **this.style.cursor =** 'move'**;**
32. **trToBeMoved = this.parentNode;**
33. **trToBeMoved.style.backgroundColor =** '#ffcccc'**;**
34. **};**
35. **td.onmouseover = function(){**//当鼠标按下，且移动到其它行上时，两行的位置进行交换
36. **if(trToBeMoved && trToBeMoved!=this.parentNode){**
37. **if(trToBeMoved.nextSibling == this.parentNode){**//往下移动
38. **trToBeMoved.parentNode.**insertBefore**(this.parentNode, trToBeMoved);**
39. **}else if(trToBeMoved.previousSibling == this.parentNode){** //往上移动
40. **trToBeMoved.parentNode.**insertBefore**(trToBeMoved, this.parentNode);**
41. **}**
42. **}**
43. **}**
44. **td.onmouseup = function(){**//当鼠标松开时，停止移动行
45. **this.style.cursor =** 'pointer'**;**
46. **if(trToBeMoved){**
47. **trToBeMoved = null;**
48. **}**
49. **};**
50. **}**
51. **}**
52. **}**
53. **}**

**步骤四：由于拖拽操作默认会选择表中的文本，影响用户拖拽排序的体验，设置表中的数据禁止选取**

修改loadData()函数，添加下述着重强调的代码：

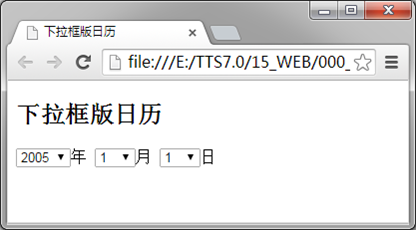
1. // 可以加载任意规整格式数据的表格
2. **function** loadData**(**data**)** **{**
3. **if** **(**data**)** **{**
4. **var** table **=** document**.**getElementById**(**'dataTable'**);**
5. clear**(**table**)** // 只要实参中有数据，则删除表中原有内容
6. createTHead**(**table**,** data**);** // 创建第一行，即表头
7. createTBody**(**table**,** data**.**rows**);** // 创建表中的记录行
8. **}**
10. //进制表格中的文本内容被选中，否则影响拖拽和再排序的效果
11. **if (typeof(table.onselectstart) !=** "undefined"**) {**
12. // IE下禁止元素被选取
13. **table.onselectstart = new Function(**"return false"**);**
14. **} else {**
15. // Firefox下不支持onselectstart，禁止元素被选取的变通办法
16. **table.onmousedown = new Function(**"return false"**);**
17. **table.onmouseup = new Function(**"return true"**);**
18. **}**
19. **}**

exercise03\_04.js文件中的完整内容如下所示：

1. //模拟从服务器获取到的具有特定格式的JSON数据
2. **var** data **=** **{**
3. "colNames" **:** **[** '订单编号'**,** '订单日期'**,** '订单总额'**,** '收货人'**,** '订单状态' **],**
4. "rows" **:** **[** **[** **9319081,** '2015-01-16 11:04'**,** **3200,** '张大力'**,** '已签收' **],**
5. **[** **9319082,** '2015-01-15 12:22'**,** **4300,** '王小玲'**,** '已签收' **],**
6. **[** **9319083,** '2015-01-14 13:12'**,** **98.5,** '张大力'**,** '派货中' **],**
7. **[** **9319084,** '2015-01-13 14:52'**,** **128.3,** '张大力'**,** '已签收' **],**
8. **[** **9319085,** '2015-01-12 15:10'**,** **33.5,** '张大力'**,** '派货中' **],**
9. **[** **9319086,** '2015-01-11 16:77'**,** **9800,** '王小玲'**,** '待付款' **]** **]**
10. **};**
11. // 可以加载任意规整格式数据的表格
12. **function** loadData**(**data**)** **{**
13. **if** **(**data**)** **{**
14. **var** table **=** document**.**getElementById**(**'dataTable'**);**
15. clear**(**table**)** // 只要实参中有数据，则删除表中原有内容
16. createTHead**(**table**,** data**);** // 创建第一行，即表头
17. createTBody**(**table**,** data**.**rows**);** // 创建表中的记录行
18. **}**
20. //进制表格中的文本内容被选中，否则影响拖拽和再排序的效果
21. **if** **(typeof(**table**.**onselectstart**)** **!=** "undefined"**)** **{**
22. // IE下禁止元素被选取
23. table**.**onselectstart **=** **new** Function**(**"return false"**);**
24. **}** **else** **{**
25. // Firefox下不支持onselectstart，禁止元素被选取的变通办法
26. table**.**onmousedown **=** **new** Function**(**"return false"**);**
27. table**.**onmouseup **=** **new** Function**(**"return true"**);**
28. **}**
29. **}**
30. // 清空表中原有内容
31. **function** clear**(**table**)** **{**
32. **while** **(**table**.**firstChild**)** **{**
33. table**.**removeChild**(**table**.**firstChild**);**
34. **}**
35. **}**
36. // 创建表头部
37. **function** createTHead**(**table**,** data**)** **{**
38. // 若有thead，则返回；否则创建新的thead并添加到table
39. **var** thead **=** table**.**createTHead**();**
40. **var** tr **=** thead**.**insertRow**();**
41. **for** **(var** i **=** **0;** i **<** data**.**colNames**.**length**;** i**++)** **{**
42. **var** td **=** document**.**createElement**(**'th'**);**// tr.insertRow()只能创建td无法创建th
43. td**.**innerHTML **=** data**.**colNames**[**i**];**
44. td**.**id **=** 'col' **+** i**;** // 为每列添加一个id属性
45. tr**.**appendChild**(**td**);**
46. **}**
47. //若data中指定了需要进行排序的列序号及排序方式
48. **if(**data**.**asc**){**
49. **var** sortedTd **=** tr**.**childNodes**[**data**.**sortedBy**];**
50. sortedTd**.**innerHTML **=** sortedTd**.**innerHTML**+(**data**.**asc**==1?**'▲'**:**'▼'**);**
51. **}**
52. // 为所有的tr增加“点击排序”功能
53. // 注意此处的技巧：没有单独为每个td添加单击处理函数，而是为整个tr添加了单击事件处理函数
54. tr**.**style**.**cursor **=** 'pointer'**;**
55. tr**.**onclick **=** **function(**event**)** **{**
56. event **=** event **||** window**.**event**;**
57. **var** td **=** event**.**srcElement **||** event**.**target**;**
58. **var** index **=** td**.**id**.**substring**(3);** //td的id值形如“col3”，index即为被点选的列在原始数据中的顺序
59. **var** sortedData **=** cloneObject**(**data**);**//将data进行深度复制，得到完全相同的一个副本
60. sortedData**.**sortedBy **=** index**;**    //接下来的排序要以哪一列为标准
61. **if(**td**.**innerHTML**.**charAt**(**td**.**innerHTML**.**length**-1)==**'▲'**){**    //若指定的列名以▲结尾
62. sortedData**.**asc **=** **-1;**    //-1表降序排列
63. **}else{**
64. sortedData**.**asc **=** **1;**        //1表升序排列
65. **}**
66. sortedData**.**rows**.**sort**(function(**row1**,** row2**)** **{**
67. **if(** row1**[**index**]** **>** row2**[**index**]){**
68. **return** **1\***sortedData**.**asc**;**
69. **}else** **if(** row1**[**index**]** **<** row2**[**index**]){**
70. **return** **-1\***sortedData**.**asc**;**
71. **}else** **{**
72. **return** **0;**
73. **}**
74. **});**
75. loadData**(**sortedData**);**
76. **}**
77. **}**
78. //对象的深度复制
79. **function** cloneObject**(**obj**)** **{**
80. **var** o **=** obj**.**constructor **===** Array **?** **[]** **:** **{};**
81. **for** **(** **var** i **in** obj**)** **{**
82. **if** **(**obj**.**hasOwnProperty**(**i**))** **{**
83. o**[**i**]** **=** **typeof** obj**[**i**]** **===** "object" **?** cloneObject**(**obj**[**i**])** **:** obj**[**i**];**
84. **}**
85. **}**
86. **return** o**;**
87. **}**
88. // 创建表体中的数据
89. **function** createTBody**(**table**,** rows**)** **{**
90. **var** tbody **=** document**.**createElement**(**'tbody'**);**// table.createTBody()IE不支持
91. table**.**appendChild**(**tbody**);**
92. **for** **(var** i **=** **0;** i **<** rows**.**length**;** i**++)** **{**
93. // 在tbody中添加一个tr
94. **var** tr **=** tbody**.**insertRow**();**
95. // 隔行变色
96. **if** **(**i **%** **2** **==** **1)** **{**
97. tr**.**className **=** 'even'**;**
98. **}**
99. **var** oldColor **=** **null;**
100. // 鼠标悬停着重强调
101. tr**.**onmouseover **=** **function()** **{**
102. oldColor **=** **this.**style**.**backgroundColor**;** // 保存原始颜色
103. **this.**style**.**backgroundColor **=** '#dddddd'**;**
104. **};**
105. // 鼠标离开恢复原色
106. tr**.**onmouseout **=** **function()** **{**
107. **this.**style**.**backgroundColor **=** oldColor**;** // 恢复原始颜色
108. **}**
109. //将要进行移动的tr
110. **var** trToBeMoved **=** **null;**
111. // 向tr中添加td
112. **for** **(var** j **=** **0;** j **<** rows**[**i**].**length**;** j**++)** **{**
113. **var** td **=** tr**.**insertCell**();**
114. td**.**innerHTML **=** rows**[**i**][**j**];**
115. **if(**j**==0){**    //当用户拖拽第0列时，可以对记录行进行重新排序
116. td**.**style**.**cursor **=** 'pointer'**;**
117. td**.**onmousedown **=** **function(){**    //当鼠标按下时，记录当前行即为要进行移动的行
118. **this.**style**.**cursor **=** 'move'**;**
119. trToBeMoved **=** **this.**parentNode**;**
120. trToBeMoved**.**style**.**backgroundColor **=** '#ffcccc'**;**
121. **};**
122. td**.**onmouseover **=** **function(){**    //当鼠标按下，且移动到其它行上时，两行的位置进行交换
123. **if(**trToBeMoved **&&** trToBeMoved**!=this.**parentNode**){**
124. **if(**trToBeMoved**.**nextSibling **==** **this.**parentNode**){**    //往下移动
125. trToBeMoved**.**parentNode**.**insertBefore**(this.**parentNode**,** trToBeMoved**);**
126. **}else** **if(**trToBeMoved**.**previousSibling **==** **this.**parentNode**){** //往上移动
127. trToBeMoved**.**parentNode**.**insertBefore**(**trToBeMoved**,** **this.**parentNode**);**
128. **}**
129. **}**
130. **}**
131. td**.**onmouseup **=** **function(){**    //当鼠标松开时，停止移动行
132. **this.**style**.**cursor **=** 'pointer'**;**
133. **if(**trToBeMoved**){**
134. trToBeMoved **=** **null;**
135. **}**
136. **};**
137. **}**
138. **}**
139. **}**
140. **}**

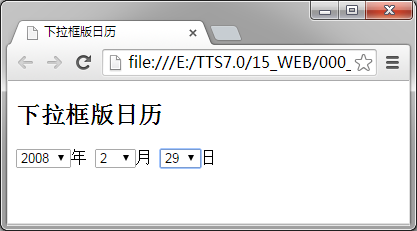
**5 下拉框版日历**

制作下拉框版的日历，页面加载后的显示效果如图－8所示：



图－8

图－8所示的页面中，年份下拉框中的数值，从2005年开始显示到当前年份；月份下拉框中的数值显示1到12；日期下拉框中的天数需要根据用户所选择的年份以及月份显示正确的天数。比如，年份如果选择为 2008，月份如果选择为 2，则最后一个下拉框中的最大天数应该是 29，此时，页面效果如图－9所示。



图－9

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_05.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**下拉框版日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面创建 js 文件**

创建名为 exercise03\_05.js 的文件，用于为页面定义 javascript 代码，然后修改上一步中所创建的 HTML 文件，为其引入 js 和样式文件。

修改后的 html 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**下拉框版日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script language=**"javascript" **src=**"exercise03\_05.js" **type=**"text/javascript"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **</**body**>**
10. **</**html**>**

**步骤三：为页面添加文本和下拉框**

为 <body> 元素添加文本和下拉框，以创建如图－8所示的页面内容。

因为年月日下拉框中的选项均由代码动态显示，因此不需要在 HTML 中书写各下拉框的选项。为便于在 js 代码中操作各下拉框，为各个 <select> 元素定义 id 属性。修改后的 html 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**下拉框版日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise03\_05.js" type**=**"text/javascript"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<h2>2.下拉框版日历</h2>**
10. **<select id=**"selYear"**></select>年**
11. **<select id=**"selMonth"**></select>月**
12. **<select id=**"selDate"**></select>日**
13. **</**body**>**
14. **</**html**>**

**步骤四：为页面元素定义事件**

上个步骤中，没有在 <select> 元素中定义 <option> 元素，而是将由代码动态的为各下拉框添加选项，因此需要为 <body> 元素定义 onload 事件，用于初始化各下拉框的选项；而当年、月的选择改变后，表示日期的下拉框中的天数会发生变化，因此还需要为表示年和月的 <select> 元素定义 onchange 事件。修改后的 HTML 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**下拉框版日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise03\_05.js" type**=**"text/javascript"**></**script**>**
7. **</**head**>**
8. **<body onload=**"initialCalendar();"**>**
9. **<**h2**>2.**下拉框版日历**</**h2**>**
10. **<select id=**"selYear" **onchange=**"dateChanged();"**></select>年**
11. **<select id=**"selMonth" **onchange=**"dateChanged();"**></select>月**
12. **<**select id**=**"selDate"**></**select**>**日
13. **</**body**>**
14. **</**html**>**

**步骤五：在 js 文件中定义方法 initialCalendar**

打开文件exercise03\_05.js，在其中添加代码，定义方法 initialCalendar，用于初始化年和月的下拉框中的选项。

假设年份从 2005 年开始，显示到当前年份（比如2014年）；而月份则显示为从1月到12月；因为默认为 1月份，则表示日期的下拉框中，显示 31 天。

js文件中添加的代码如下所示：

1. //初始化日历的相关择框
2. **function** initialCalendar**()** **{**
3. //定义开始年份
4. **var** startYear **=** **2005;**
5. //声明 option 对象的索引
6. **var** index **=** **0;**
7. //初始化年份选择框
8. **var** yearObj **=** document**.**getElementById**(**"selYear"**);**
9. **var** endYear **=** **(new** Date**()).**getFullYear**();**
10. **for** **(var** i **=** startYear**;** i **<=** endYear**;** i**++)** **{**
11. **var** optionObj **=** **new** Option**(**i**,** i**);**
12. yearObj**.**options**[**index**]** **=** optionObj**;**
13. index**++;**
14. **}**
15. //初始化月份选择框
16. **var** monthObj **=** document**.**getElementById**(**"selMonth"**);**
17. index **=** **0;**
18. **for** **(var** i **=** **1;** i **<=** **12;** i**++)** **{**
19. **var** optionObj **=** **new** Option**(**i**,** i**);**
20. monthObj**.**options**[**index**]** **=** optionObj**;**
21. index**++;**
22. **}**
23. //初始化日期选择框
24. **var** dateObj **=** document**.**getElementById**(**"selDate"**);**
25. index **=** **0;**
26. **for** **(var** i **=** **1;** i **<=** **31;** i**++)** **{**
27. **var** optionObj **=** **new** Option**(**i**,** i**);**
28. dateObj**.**options**[**index**]** **=** optionObj**;**
29. index**++;**
30. **}**
31. **}**

**步骤六：在 js 文件中定义方法 getDays**

当年份和月份的选择发生变化后，首先需要判断该年该月下的天数，然后才能为日期下拉框动态的创建天数选项。

因此，继续为文件exercise03\_05.js添加代码，定义名为 getDays的方法，用来计算某年某月的天数。需要注意的是，闰年的 2 月份有29天。

js文件中添加的代码如下所示：

1. //获得某年某月的总天数
2. **function** getDays**(**year**,** month**)** **{**
3. **if** **(**month **==** **2)** **{**
4. **var** isLeap **=** **(**year **%** **4** **==** **0** **&&** year **%** **100** **!=** **0)** **||** year **%** **400** **==** **0;**
5. **return** isLeap **?** **29** **:** **28;**
6. **}**
7. **else** **if** **(**month **==** **4** **||** month **==** **6** **||** month **==** **9** **||** month **==** **11)**
8. **return** **30;**
9. **else**
10. **return** **31;**
11. **}**

**步骤七：在 js 文件中定义方法 dateChanged**

继续为文件exercise03\_05.js 添加代码，定义名为 dateChanged的方法，用于根据年份和月份的选择，动态的创建日期选项。代码如下所示：

1. //年份和月份选择改变后，修改日期下拉框
2. **function** dateChanged**()** **{**
3. //得到所选择的年份
4. **var** yearObj **=** document**.**getElementById**(**"selYear"**);**
5. **var** year **=** yearObj**.**options**[**yearObj**.**selectedIndex**].**value**;**
6. //得到所选择的月份
7. **var** monthObj **=** document**.**getElementById**(**"selMonth"**);**
8. **var** month **=** monthObj**.**options**[**monthObj**.**selectedIndex**].**value**;**
9. //得到当月最大天数
10. **var** days **=** getDays**(**year**,** month**);**
11. //删除原有天数
12. **var** dateObj **=** document**.**getElementById**(**"selDate"**);**
13. dateObj**.**options**.**length **=** **0;**
14. //重新添加天
15. **var** index **=** **0;**
16. **for** **(var** i **=** **1;** i **<=** days**;** i**++)** **{**
17. **var** optionObj **=** **new** Option**(**i**,** i**);**
18. dateObj**.**options**[**index**]** **=** optionObj**;**
19. index**++;**
20. **}**
21. **}**

exercise03\_05.html 文件中的完整代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**下拉框版日历**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise03\_05.js" type**=**"text/javascript"**></**script**>**
7. **</**head**>**
8. **<**body onload**=**"initialCalendar();"**>**
9. **<**h2**>2.**下拉框版日历**</**h2**>**
10. **<**select id**=**"selYear" onchange**=**"dateChanged();"**></**select**>**年
11. **<**select id**=**"selMonth" onchange**=**"dateChanged();"**></**select**>**月
12. **<**select id**=**"selDate"**></**select**>**日
13. **</**body**>**
14. **</**html**>**

exercise03\_05.js 文件中的完整代码如下所示：

1. //初始化日历的相关择框
2. **function** initialCalendar**()** **{**
3. //定义开始年份
4. **var** startYear **=** **2005;**
5. //声明 option 对象的索引
6. **var** index **=** **0;**
7. //初始化年份选择框
8. **var** yearObj **=** document**.**getElementById**(**"selYear"**);**
9. **var** endYear **=** **(new** Date**()).**getFullYear**();**
10. **for** **(var** i **=** startYear**;** i **<=** endYear**;** i**++)** **{**
11. **var** optionObj **=** **new** Option**(**i**,** i**);**
12. yearObj**.**options**[**index**]** **=** optionObj**;**
13. index**++;**
14. **}**
15. //初始化月份选择框
16. **var** monthObj **=** document**.**getElementById**(**"selMonth"**);**
17. index **=** **0;**
18. **for** **(var** i **=** **1;** i **<=** **12;** i**++)** **{**
19. **var** optionObj **=** **new** Option**(**i**,** i**);**
20. monthObj**.**options**[**index**]** **=** optionObj**;**
21. index**++;**
22. **}**
23. //初始化日期选择框
24. **var** dateObj **=** document**.**getElementById**(**"selDate"**);**
25. index **=** **0;**
26. **for** **(var** i **=** **1;** i **<=** **31;** i**++)** **{**
27. **var** optionObj **=** **new** Option**(**i**,** i**);**
28. dateObj**.**options**[**index**]** **=** optionObj**;**
29. index**++;**
30. **}**
31. **}**
32. //年份和月份选择改变后，修改日期下拉框
33. **function** dateChanged**()** **{**
34. //得到所选择的年份
35. **var** yearObj **=** document**.**getElementById**(**"selYear"**);**
36. **var** year **=** yearObj**.**options**[**yearObj**.**selectedIndex**].**value**;**
37. //得到所选择的月份
38. **var** monthObj **=** document**.**getElementById**(**"selMonth"**);**
39. **var** month **=** monthObj**.**options**[**monthObj**.**selectedIndex**].**value**;**
40. //得到当月最大天数
41. **var** days **=** getDays**(**year**,** month**);**
42. //删除原有天数
43. **var** dateObj **=** document**.**getElementById**(**"selDate"**);**
44. dateObj**.**options**.**length **=** **0;**
45. //重新添加天
46. **var** index **=** **0;**
47. **for** **(var** i **=** **1;** i **<=** days**;** i**++)** **{**
48. **var** optionObj **=** **new** Option**(**i**,** i**);**
49. dateObj**.**options**[**index**]** **=** optionObj**;**
50. index**++;**
51. **}**
52. **}**
53. //获得某年某月的总天数
54. **function** getDays**(**year**,** month**)** **{**
55. **if** **(**month **==** **2)** **{**
56. **var** isLeap **=** **(**year **%** **4** **==** **0** **&&** year **%** **100** **!=** **0)** **||** year **%** **400** **==** **0;**
57. **return** isLeap **?** **29** **:** **28;**
58. **}**
59. **else** **if** **(**month **==** **4** **||** month **==** **6** **||** month **==** **9** **||** month **==** **11)**
60. **return** **30;**
61. **else**
62. **return** **31;**
63. **}**

**1 列举BOM对象及各自的含义**

**参考答案**

BOM：Browser Object Model，使JavaScript可以与浏览器窗口本身进行交互。注意：BOM模型没有相关行业标准，常用功能被大部分浏览器所支持，但有些实现细节上的不同也是造成js浏览器兼容性问题的重要原因。它包含如下对象：

window：表示打开的浏览器窗口

screen：包含当前客户端屏幕的信息

navigator：包含浏览器应用程序的信息

location：包含当前显示的URL的相关信息

history：包含浏览历史记录信息

document：表示浏览器当前加载的文档

event：包含当前用户操作所触发的事件的信息

上述BOM对象中，window对象是根对象，其它对象都是window对象的成员属性。例如screen是window.screen的缩写。

**2 闪烁的标题栏**

有些页面为了引起用户的注意，特意将浏览器标题栏中的内容实现“动画”效果，如不停的改变标题栏中显示的内容，如图-1所示：



图-1

大约0.5秒钟后，标题栏内容变为图-2所示：

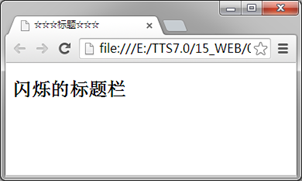


图-2

依此往复，从而实现引起用户注意的目的。试实现此效果。

**参考答案**

此案例需要调用修改文档标题栏的方法，且需要使用定时器，每隔500毫秒即调用一次，来实现类似“动画”的效果。使用类似的原理，也可以实现类似“跑马灯”等更多的动画效果。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**闪烁的标题栏**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **<**h2**>**闪烁的标题栏**</**h2**>**
9. **</**body**>**
10. **</**html**>**

**步骤二：添加外部JavaScript引用，当文档加载完成时调用其中的方法**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**闪烁的标题栏**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**src=**"exercise04\_02.js"**></script>**
7. **</**head**>**
8. **<**body onload**=**"flashTitle()"**>**
9. **<**h2**>**闪烁的标题栏**</**h2**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：创建外部JavaScript文件exercise04\_02.js,编写flashTitle()方法**

1. /\*闪动的标题栏\*/
2. **function** flashTitle**(){**
3. //每隔500毫秒修改一次标题栏内容
4. setInterval**(**'setTitle()'**,** **500);**
5. **}**
6. /\*全局函数，用于记录闪烁的次数\*/
7. **var** count **=** **0;**
8. /\*修改文档显示的标题\*/
9. **function** setTitle**(){**
10. **if(**count**==0){**
11. count**=1;**
12. document**.**title **=** '☆☆☆标题☆☆☆'**;**
13. **}else{**
14. count**=0;**
15. document**.**title **=** '★★★标题★★★'**;**
16. **}**
17. **}**

**3 页面内进度条效果**

在页面中模拟实现如图-3中的效果：

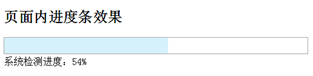


图-3

图中的进度条以随机的速度前进，直至100%。

**参考答案**

每隔一段时间，上图中的进度条就会前进一段随机的距离，这种功能可以使用定时器加以模拟实现。由于每隔一定时间执行一次，可以使用setInterval()来实现，在进度条到100后取消定时器；也可以使用setTimeout()执行一次进度条的前进，若发现没有到100，则在执行一次当前函数——递归调用，下面的实现采用了第二种方案。

具体过程如下：

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_03.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**页面内进度条效果**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript和CSS文件引用**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**页面内进度条效果**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise04\_03.css"**/>**
7. **<script type=**"text/javascript"**src=**"exercise04\_03.js"**></script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：添加HTML元素，以及页面加载完成后调用的方法**

注意：为了避免空白字符被识别为子节点问题，进度条div中的内容(span和div)写在了同一行中。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**页面内进度条效果**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**link rel**=**"stylesheet" type**=**"text/css" href**=**"exercise04\_03.css"**/>**
7. **<**script type**=**"text/javascript"src**=**"exercise04\_03.js"**></**script**>**
8. **</**head**>**
9. **<**body onload**=**"progress()"**>**
10. **<h2>页面内进度条效果</h2>**
11. **<div id=**"bar"**><div><**/div><span>系统检测进度：0%</**span></div>**
12. **</**body**>**
13. **</**html**>**

**步骤四：创建外部CSS文件exercise04\_03.css**

1. #bar**{**    /\*进度条\*/
2. background**-**color**:** #ffffff**;**
3. border**:** 1px solid #aaaaaa**;**
4. width**:** 500px**;**
5. height**:** 26px**;**
6. **}**
7. #bar div**{**    /\*显示进度的部分\*/
8. background**-**color**:** #D5F1FC**;**
9. height**:** 26px**;**
10. width**:** **0;**
11. **}**
12. #bar span**{** /\*进度条内的提示文字\*/
13. line**-**height**:** 26px**;**
14. **}**

**步骤五：创建外部JavaScript文件exercise04\_03.js**

1. /\*进度值: 0-100\*/
2. **var** value**=0;**
3. /\*修改进度条进度\*/
4. **function** progress**(){**
5. **var** bar **=** document**.**getElementById**(**'bar'**);**
6. **var** div **=** bar**.**firstChild**;**    //进度条内部显示进度的div
7. **var** span **=** bar**.**lastChild**;**    //进度条内部显示文字的span
9. //获取元素内嵌样式中的宽度可使用 e.style.width
10. //但对于获取外部样式文件中指定的宽度必须考虑浏览器兼容性问题
11. **var** barWidth **=** **null;**    //整个进度条外框的最终显示样式对象中的宽度
12. **if(**bar**.**currentStyle**){**    //IE使用 e.currentStyle获取元素样式对象
13. barWidth **=** bar**.**currentStyle**.**width**;**
14. **}else{**                    //非IE使用window.getComputedStyle(e, styleName)获取元素样式对象
15. barWidth **=** window**.**getComputedStyle**(**bar**,null).**width**;**
16. **}**
17. barWidth **=** parseInt**(**barWidth**);**
19. //修改进度条宽度和提示文本内容
20. div**.**style**.**width **=** **(**value**/100)\***barWidth**+**'px'**;**
21. span**.**innerHTML **=** '系统检测进度：'**+**value**+**'%'**;**

24. //若进度值未到100，则继续调用修改进度方法
25. **if(**value**<100){**
26. setTimeout**(**'progress()'**,** **1000);** //递归调用
27. **}**
29. //修改进度值
30. value **+=** **(**parseInt**(**Math**.**random**()\*20)** **+** **10);**
31. value **=** value**>100** **?** **100** **:** value**;**
32. **}**

**4 广告轮播（提高题，选做）**

有四张图片，其名称和图片效果如图－4所示：



图－4

需要使用图－4中的四张图片完成页面上的广告轮播效果，详细要求如下：

1、页面加载后，先在页面上显示第一张图片（rose.jpg），当前图片所对应的数字编号也高亮显示。页面效果如图－5所示。



图－5

2、每隔3s，自动更换页面上的图片显示，并修改图片所对应的数字编号的样式，效果如图－6所示：



图－6

3、图片轮换显示的顺序为："rose.jpg"、"daffodil.jpg"、"apple.jpg"、 "daisy.jpg"；

4、一轮显示完毕后，重复下一轮显示；

5、鼠标移入图片时，停止图片轮换，鼠标移出图片时，继续图片轮换；

6、鼠标移入图片下方的某个数字编号，则显示该数字所对应的图片，且停止图片轮换；

7、鼠标移出图片下方的某个数字编号，则继续图片轮换。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_04.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**广告轮播**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面创建 js 文件和 CSS 文件**

创建名为 exercise04\_04.js 的文件，用于为页面定义 javascript 代码；并创建名为 exercise04\_04.css 的样式文件。然后修改上一步中所创建的 HTML 文件，为其引入 js 和样式文件。

修改后的 html 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**广告轮播**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script language=**"javascript" **src=**"exercise03\_03.js" **type=**"text/javascript"**></script>**
7. **<link href=**"exercise03\_03.css" **type=**"text/css" **rel=**"Stylesheet" **/>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：为页面添加图片和数字编号**

为 <body> 元素添加图片以及数字编号，以创建如图－7所示的页面内容（初始显示第一张图片）。修改后的 html 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**广告轮播**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise04\_04.js" type**=**"text/javascript"**></**script**>**
7. **<**link href**=**"exercise04\_04.css" type**=**"text/css" rel**=**"Stylesheet" **/>**
8. **</**head**>**
9. **<**body**>**
10. **<h2>广告轮播</h2>**
11. **<img id=**"img1" **src=**"img/rose.jpg" **/>**
12. **<div id=**"indexDiv"**>**
13. **<span class=**"current"**>1</span>**
14. **<span>2</span>**
15. **<span>3</span>**
16. **<span>4</span>**
17. **</div>**
18. **</**body**>**
19. **</**html**>**

需要注意的是，为便于样式定义以及动态的操作页面元素，为 <img> 定义了 id 属性，为包含数字的 <div> 元素也定义了 id 属性；另外，为了突出显示当前的数字，为其定义 class 属性，用于设置样式。

**步骤四：为页面定义样式**

打开 exercise04\_04.css 文件，为页面定义样式。

首先为 <img> 定义样式：设置图片的大小；然后为包含数字的 <div> 定义样式：设置高度、宽度以及字体大小。CSS 文件中添加的代码如下所示：

1. img
2. **{**
3. width**:**200px**;**
4. **}**
5. div
6. **{**
7. height**:**50px**;**
8. width**:**200px**;**
9. font**-**size**:**12pt**;**
10. **}**

**步骤五：为数字文本定义样式**

继续为文件 exercise04\_04.css 添加代码，为页面上的数字文本定义样式，设置其背景色、边距以及光标等。并定义样式类 current 用于定义当前数字的样式：背景色为红色。CSS 文件中添加的代码如下所示：

1. div span
2. **{**
3. padding**:**8px 10px**;**
4. background**-**color**:**#dddddd**;**
5. margin**-**right**:**15px**;**
6. width**:**200px**;**
7. line**-**height**:**50px**;**
8. cursor**:**pointer**;**
9. **}**
10. span**.**current
11. **{**
12. background**-**color**:**#f00**;**
13. **}**

**步骤六：为页面元素定义事件**

页面加载后，就出现图片轮换，因此，需要为 <body> 元素定义 onload 事件；而鼠标移入图片和数字，则停止轮换，鼠标移出图片和数字，继续轮换，因此，还需要为 <img> 元素和 <span> 元素添加鼠标移入和鼠标移出事件的定义。修改后的 HTML 代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**广告轮播**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise04\_04.js" type**=**"text/javascript"**></**script**>**
7. **<**link href**=**"exercise04\_04.css" type**=**"text/css" rel**=**"Stylesheet" **/>**
8. **</**head**>**
9. **<body onload=**"startRotate();"**>**
10. **<**h2**>**广告轮播**</**h2**>**
11. **<img id=**"img1" **src=**"img/rose.jpg" **onmouseover=**"stopRotate();" **onmouseout=**"startRotate();" **/>**
12. **<**div id**=**"indexDiv"**>**
13. **<span class=**"current" **onmouseover=**"stopRotate(0);"
14. **onmouseout=**"startRotate();"**>1</span>**
15. **<span onmouseover=**"stopRotate(1);"
16. **onmouseout=**"startRotate();"**>2</span>**
17. **<span onmouseover=**"stopRotate(2);"
18. **onmouseout=**"startRotate();"**>3</span>**
19. **<span onmouseover=**"stopRotate(3);"
20. **onmouseout=**"startRotate();"**>4</span>**
21. **</**div**>**
22. **</**body**>**
23. **</**html**>**

需要注意的是，因为当鼠标移入某数字的时候，需要立即显示当前数字所对应的图片，因此，需要将显示的图片序号作为参数传入方法。

**步骤七：在 js 文件中定义全局变量**

打开文件exercise04\_04.js，在其中添加代码，定义所需要的全局变量。

首先，创建一个数组，用于存储所需要显示的图片的路径；然后定义变量用于表示即将显示的下一张图片在数组中的索引；最后定义变量表示被启动的定时器对象。

js文件中添加的代码如下所示：

1. //数组：存储图片路径
2. **var** imageArray **=** **[**"img/rose.jpg"**,** "img/daffodil.jpg"**,** "img/apple.jpg"**,** "img/daisy.jpg"**];**
3. //即将显示的下一张图片在数组中的索引，因为页面初始化显示第一张图片，因此，下一张图片为第二张图片
4. **var** imageIndex **=** **1;**
5. //定时器
6. **var** imageTimer**;**

**步骤八：在 js 文件中定义方法 startRotate**

继续为文件exercise04\_04.js添加代码，定义名为 startRotate的方法，用来启动定时器以修改页面显示的图片以及数字的样式。因为所启动的相关功能比较复杂，代码也比较多，单独定义一个方法modifyImageAndSpan 来实现。

js文件中添加的代码如下所示：

1. //启动图片轮换
2. **function** startRotate**()** **{**
3. //启动定时器，每 3 s 修改一次：3s 后修改时，直接显示为下一张图片
4. imageTimer **=** window**.**setInterval**(**modifyImageAndSpan**,** **3000);**
5. **}**

**步骤九：在 js 文件中定义方法 modifyImageAndSpan**

继续为文件exercise04\_04.js 添加代码，定义名为 modifyImageAndSpan的方法，用于修改图片显示以及修改数字文本的样式。代码如下所示：

1. /\*改变图片以及对应序号的样式：
2. 1、先根据当前应该显示的图片序号（imageIndex）更新图片和数字文本的样式
3. 2、再更新图片序号（为下一次显示做准备）\*/
4. **function** modifyImageAndSpan**()** **{**
5. //找到 <img>元素，修改其图片为当前应该显示的图片（下一张图片）
6. **var** image **=** document**.**getElementById**(**"img1"**);**
7. image**.**src **=** imageArray**[**imageIndex**];**
8. //修改<span>元素的样式：只有和图片一致的数字显示为 current 样式
9. **var** div **=** document**.**getElementById**(**"indexDiv"**);**
10. **var** spanNodes **=** div**.**getElementsByTagName**(**"span"**);**
11. **for** **(var** i **=** **0;** i **<** spanNodes**.**length**;** i**++)** **{**
12. **if** **(**i **==** imageIndex**)**
13. spanNodes**[**i**].**className **=** "current"**;**
14. **else**
15. spanNodes**[**i**].**removeAttribute**(**"class"**);**
16. **}**
17. //更新图片索引，用于下一次的图片显示：如果已经是最后一张图片，则重新开始
18. **if** **(**imageIndex **==** imageArray**.**length **-** **1)**
19. imageIndex **=** **0;**
20. **else**
21. imageIndex**++;**
22. **}**

**步骤十：在 js 文件中定义方法 stopRotate**

继续为文件exercise04\_04.js 添加代码，定义名为 stopRotate的方法，用于停止定时器。

有两种动作可以停止定时器：鼠标移入图片和鼠标移入数字文本。鼠标移入数字文本时，会传入所需要即时显示的图片的序号，因此，在方法stopRotate中，需要判断是否有参数传入。代码如下所示：

1. //停止图片轮换
2. **function** stopRotate**()** **{**
3. //先判断是否有参数：有参数则根据参数更新 imageIndex，并立即切换到相应的图片
4. **if** **(**arguments**.**length **==** **1)** **{**
5. imageIndex **=** arguments**[0];**
6. modifyImageAndSpan**();**
7. **}**
8. //停止定时器
9. window**.**clearInterval**(**imageTimer**);**
10. **}**

exercise04\_04.html 文件中的完整代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**广告轮播**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script language**=**"javascript" src**=**"exercise04\_04.js" type**=**"text/javascript"**></**script**>**
7. **<**link href**=**"exercise04\_04.css" type**=**"text/css" rel**=**"Stylesheet" **/>**
8. **</**head**>**
9. **<**body onload**=**"startRotate();"**>**
10. **<**form**>**
11. **<**h2**>**广告轮播**</**h2**>**
12. **<**img id**=**"img1" src**=**"img/rose.jpg" onmouseover**=**"stopRotate();" onmouseout**=**"startRotate();" **/>**
13. **<**div id**=**"indexDiv"**>**
14. **<**span **class=**"current" onmouseover**=**"stopRotate(0);"
15. onmouseout**=**"startRotate();"**>1</**span**>**
16. **<**span onmouseover**=**"stopRotate(1);"
17. onmouseout**=**"startRotate();"**>2</**span**>**
18. **<**span onmouseover**=**"stopRotate(2);"
19. onmouseout**=**"startRotate();"**>3</**span**>**
20. **<**span onmouseover**=**"stopRotate(3);"
21. onmouseout**=**"startRotate();"**>4</**span**>**
22. **</**div**>**
23. **</**form**>**
24. **</**body**>**
25. **</**html**>**

exercise04\_04.css 文件中的完整代码如下所示：

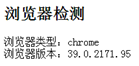
1. img
2. **{**
3. width**:**200px**;**
4. **}**
5. div
6. **{**
7. height**:**50px**;**
8. width**:**200px**;**
9. font**-**size**:**12pt**;**
10. **}**
11. div span
12. **{**
13. padding**:**8px 10px**;**
14. background**-**color**:**#dddddd**;**
15. margin**-**right**:**15px**;**
16. width**:**200px**;**
17. line**-**height**:**50px**;**
18. cursor**:**pointer**;**
19. **}**
20. span**.**current
21. **{**
22. background**-**color**:**#f00**;**
23. **}**

exercise04\_04.js 文件中的完整代码如下所示：

1. //数组：存储图片路径
2. **var** imageArray **=**
3. **[**"img/rose.jpg"**,**
4. "img/daffodil.jpg"**,**
5. "img/apple.jpg"**,**
6. "img/daisy.jpg"**];**
7. //即将显示的下一张图片在数组中的索引，因为页面初始化显示第一张图片，因此，下一张图片为第二张图片
8. **var** imageIndex **=** **1;**
9. //定时器
10. **var** imageTimer**;**
11. /\*改变图片以及对应序号的样式：
12. 1、先根据当前应该显示的图片序号（imageIndex）更新图片和数字文本的样式
13. 2、再更新图片序号（为下一次显示做准备）\*/
14. **function** modifyImageAndSpan**()** **{**
15. //找到 <img>元素，修改其图片为当前应该显示的图片（下一张图片）
16. **var** image **=** document**.**getElementById**(**"img1"**);**
17. image**.**src **=** imageArray**[**imageIndex**];**
18. //修改<span>元素的样式：只有和图片一致的数字显示为 current 样式
19. **var** div **=** document**.**getElementById**(**"indexDiv"**);**
20. **var** spanNodes **=** div**.**getElementsByTagName**(**"span"**);**
21. **for** **(var** i **=** **0;** i **<** spanNodes**.**length**;** i**++)** **{**
22. **if** **(**i **==** imageIndex**)**
23. spanNodes**[**i**].**className **=** "current"**;**
24. **else**
25. spanNodes**[**i**].**removeAttribute**(**"class"**);**
26. **}**
27. //更新图片索引，用于下一次的图片显示：如果已经是最后一张图片，则重新开始
28. **if** **(**imageIndex **==** imageArray**.**length **-** **1)**
29. imageIndex **=** **0;**
30. **else**
31. imageIndex**++;**
32. **}**
33. //启动图片轮换
34. **function** startRotate**()** **{**
35. //启动定时器，每 3 s 修改一次：3s 后修改时，直接显示为下一张图片
36. imageTimer **=** window**.**setInterval**(**modifyImageAndSpan**,** **3000);**
37. **}**
38. //停止图片轮换
39. **function** stopRotate**()** **{**
40. //先判断是否有参数：有参数则根据参数更新 imageIndex，并立即切换到相应的图片
41. **if** **(**arguments**.**length **==** **1)** **{**
42. imageIndex **=** arguments**[0];**
43. modifyImageAndSpan**();**
44. **}**
45. //停止定时器
46. window**.**clearInterval**(**imageTimer**);**
47. **}**

**5 浏览器检测**

准确指明浏览器类型及版本号，如图-7所示：



图－7

**参考答案**

浏览器类型及版本保存在window.navigator对象中，其中信息最丰富的是navigator.userAgent属性，但此属性中数据较杂乱，需要使用正则表达式进行解析。实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_05.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**浏览器检测**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript文件引用**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**浏览器检测**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**src=**"exercise04\_05.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **</**body**>**
10. **</**html**>**

**步骤三：添加HTML元素，以及页面加载完成后调用的函数**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**浏览器检测**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"src**=**"exercise04\_05.js"**></**script**>**
7. **</**head**>**
8. **<**body onload**=**"checkBrowser()"**>**
9. **<h2>浏览器检测</h2>**
10. **<div id=**"checkingResult"**></div>**
11. **</**body**>**
12. **</**html**>**

**步骤四：创建外部JavaScript文件exercise04\_05.js**

1. /\*浏览器检测\*/
2. **function** checkBrowser**()** **{**
3. **var** browser **=** **{**
4. appname **:** 'unknown'**,**
5. version **:** **0**
6. **}**
7. **var** userAgent **=** navigator**.**userAgent**.**toLowerCase**();**
8. **if** **(**/(msie|firefox|opera|chrome|netscape)\D+(\d[\d.]\*)/**.**test**(**userAgent**))** **{**
9. browser**.**appname **=** RegExp**.**$**1;**
10. browser**.**version **=** RegExp**.**$**2;**
11. **}** **else** **if** **(**/version\D+(\d[\d.]\*).\*safari/**.**test**(**userAgent**))** **{** // safari
12. browser**.**appname **=** 'safari'**;**
13. browser**.**version **=** RegExp**.**$**2;**
14. **}**
15. // 将检测结果显示在页面中
16. **var** div **=** document**.**getElementById**(**'checkingResult'**);**
17. div**.**innerHTML **=** '浏览器类型：' **+** browser**.**appname **+** '<br/>浏览器版本：'    **+** browser**.**version**;**
18. **}**

**6 总结可以实现页面刷新的方法**

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_06.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**总结可以实现页面刷新的方法**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加可以实现页面重新加载的代码及测试按钮**

简单起见，这些代码未单列到外部文件或函数中。编写时，注意浏览器兼容性问题。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**总结可以实现页面刷新的方法**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **<h2>总结可以实现页面刷新的方法</h2>**
10. **<h3>所有浏览器都兼容的方法</h3>**
11. **<button onclick=**"history.go(0)"**>history.**go**(0)</button>**
12. **<button onclick=**"location.reload()"**>location.**reload**()</button>**
13. **<button onclick=**"location=location"**>location=location</button>**
14. **<button onclick=**"location.assign(location)"**>location.**assign**(location)</button>**
15. **<button onclick=**"location.replace(location)"**>location.**replace**(location)</button>**
17. **<h3>仅IE中可以使用的方法</h3>**
18. **<button onclick=**"document.URL=location.href"**>document.URL=location.href</button>**
19. **<button onclick=**"document.execCommand('Refresh')"**>document.**execCommand**(**'Refresh'**)</button>**
20. **<button onclick=**"window.navigate(location)"**>window.**navigate**(location)</button>**
21. **</**body**>**
22. **</**html**>**

**1 实现选择框功能**

有页面效果如图－1所示：



图－1

图－1所示的页面中，显示多个可供选择的菜名，如果用户选择了某道菜（如鱼香肉丝），则对应菜名会显示在页面下方的框中，页面效果如图－2所示：



图－2

再次点击某道菜名，则该菜在下方框中消失。

另外，如果单击下方框中某道菜右上角的“X”图片，则此道菜从“已选择”框中删除，同时上方“午间菜谱”框中对应项取消选定。如图-3所示：



图－3

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_01.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现选择框功能**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript和CSS文件引用，**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现选择框功能**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**src=**"exercise05\_01.js"**></script>**
7. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise05\_01.css"**/>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：添加HTML元素**

“午间菜谱”框每个checkbox选项必须有唯一的value值；该元素后面的菜名统一放在了一个SPAN元素内，这样便于稍后查找菜名。

页面中没有为任何元素绑定单击事件处理方法，所以需要在页面加载完成后查找到所有的复选框元素，为它们绑定单击事件处理方法。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现选择框功能**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"src**=**"exercise05\_01.js"**></**script**>**
7. **<**link rel**=**"stylesheet" type**=**"text/css" href**=**"exercise05\_01.css"**/>**
8. **</**head**>**
9. **<**body onload**=**"bind()"**>**
10. **<h2>实现选择框功能</h2>**
11. **<fieldset id=**"menu"**>**
12. **<legend>午间食谱</legend>**
13. **<input type=**"checkbox" **value=**"1" **name=**"dish"/><span>鱼香肉丝</**span>**
14. **<input type=**"checkbox" **value=**"2" **name=**"dish"/><span>京酱肉丝</**span>**
15. **<input type=**"checkbox" **value=**"3" **name=**"dish"/><span>回锅肉</**span>**
16. **<input type=**"checkbox" **value=**"4" **name=**"dish"/><span>木须肉</**span>**
17. **</fieldset>**
18. **<br/><hr/><br/>**
19. **<fieldset id=**"chosen"**>**
20. **<legend>已选择</legend>**
21. **</fieldset>**
22. **</**body**>**
23. **</**html**>**

**步骤四：创建外部CSS样式文件exercise05\_01.css**

1. /\*“所有菜谱”和“已选择”两个框的样式\*/
2. fieldset**{**
3. width**:**400px**;**
4. height**:**60px**;**
5. border**:**1px solid gray**;**
6. padding**:** 20px**;**
7. **}**
8. /\*“已选择”框中显示的一个被选定的菜\*/
9. #chosen div**{**
10. float**:**left**;**
11. height**:**30px**;**
12. line**-**height**:**30px**;**
13. border**:**1px solid gray**;**
14. background**-**color**:**#ddd**;**
15. text**-**align**:**center**;**
16. padding**:**8px**;**
17. margin**-**right**:**10px**;**
18. **}**
19. /\*被选定的菜名右上角的X号图片\*/
20. #chosen div img**{**
21. position**:** absolute**;**
22. cursor**:** pointer**;**
23. width**:**6px**;**
24. **}**

**步骤五：创建外部JavaScript文件exercise05\_01.js**

需要创建三个函数：

-bind():查找checkbox元素，绑定单击事件处理方法

-addDish(id,name)：将选定的菜添加到“已选择”框

-removeDish(id)：从“已选择”框中删除一道菜

1. /\*当页面加载完成后，为复选框绑定单击事件处理方法\*/
2. **function** bind**()** **{**
3. // 查找fieldset及其中所有checkbox input
4. **var** fieldset **=** document**.**getElementById**(**'menu'**);**
5. **var** inputArr **=** fieldset**.**getElementsByTagName**(**'input'**);**
6. // 为每个checkbox的单击事件绑定处理方法
7. **for** **(var** i **=** **0;** i **<** inputArr**.**length**;** i**++)** **{**
8. inputArr**[**i**].**onclick **=** **function()** **{** // 匿名函数
9. **if** **(this.**checked**)** **{**
10. addDish**(this.**value**,** **this.**nextSibling**.**innerHTML**);**// 添加菜名
11. **}** **else** **{**
12. removeDish**(this.**value**);**// 删除菜名
13. **}**
14. **}**
15. **}**
16. **}**
17. /\*
18. \* 添加菜名
19. \* 1、创建形如<div><input type="hidden" value="1" /><span>鱼香肉丝</span><img src="del.gif"/></div>的内容
20. \* 2、加入到“已选择”包含框中
21. \*/
22. **function** addDish**(**id**,** name**)** **{**
23. // 创建div
24. **var** div **=** document**.**createElement**(**"div"**);**
25. // 创建hidden input，用于记载 id 值
26. **var** hidden **=** document**.**createElement**(**"input"**);**
27. hidden**.**type **=** "hidden"**;**
28. hidden**.**value **=** id**;**
29. // 创建 span，用于显示菜名
30. **var** span **=** document**.**createElement**(**"span"**);**
31. span**.**innerHTML **=** name**;**
32. // 创建关闭用的图像
33. **var** img **=** document**.**createElement**(**"img"**);**
34. img**.**src **=** "img/del.gif"**;**
35. img**.**onclick **=** **function()** **{**
36. //在“菜谱”框中找到对应的复选框，取消选定
37. **var** fieldset **=** document**.**getElementById**(**'menu'**);**
38. **var** inputArr **=** fieldset**.**getElementsByTagName**(**'input'**);**
39. **for** **(var** i **=** **0;** i **<** inputArr**.**length**;** i**++)** **{**
40. **if(**inputArr**[**i**].**value **==** id**){**
41. inputArr**[**i**].**checked **=** **false;**
42. **break;**
43. **}**
44. **}**
45. //在“已选择”框中删除指定的菜
46. removeDish**(**id**);**
47. **};**
48. // 加入 div
49. div**.**appendChild**(**hidden**);**
50. div**.**appendChild**(**span**);**
51. div**.**appendChild**(**img**);**
52. // 加入“已选择”包含框
53. document**.**getElementById**(**"chosen"**).**appendChild**(**div**);**
54. **}**
55. /\* 根据id值在“已选择”框中移除菜名\*/
56. **function** removeDish**(**id**)** **{**
57. // 获取“已选择”框
58. **var** fieldset **=** document**.**getElementById**(**"chosen"**);**
59. // 找到所有的隐藏域，根据id值从FIELDSET中删除相应的 DIV元素
60. **var** hiddens **=** fieldset**.**getElementsByTagName**(**"input"**);**
61. **for** **(var** i **=** **0;** i **<** hiddens**.**length**;** i**++)** **{**
62. **if** **(**hiddens**[**i**].**value **==** id**)** **{**
63. fieldset**.**removeChild**(**hiddens**[**i**].**parentNode**);**
64. **break;**
65. **}**
66. **}**
67. **}**

**2 如何在js代码中，获取 event 对象并读取相关信息？**

**参考答案：**

可以在 HTML 页面或者 js 代码中使用 event 关键字直接获取 event 对象，但是考虑到浏览器兼容问题，建议在页面元素的事件句柄定义中，使用 event 关键字将事件对象作为参数传入 js 中的方法，代码如下所示：

1. **<**input type**=**"button" value**=**"获取 event 对象" onclick**=**"getEventData(event);" **/>**

在 js 代码中，使用参数接收 event 对象，代码如下所示：

1. **function** getEventData**(**eObj**){**
2. //参数 eObj 用于接收传入的 event 对象
3. **}**

获得 event 对象后，可以使用 event对象的相关属性获取事件的信息，其中，最重要的信息就是获取触发事件的元素对象。在IE 等浏览器中，使用 event.srcElement 属性即可获得该元素节点；而在 Firefox 浏览器中，需要使用 event.target属性获取。因此，考虑到浏览器兼容问题，往往使用如下代码获取：

1. **var** node **=** eObj**.**srcElement **||** eObj**.**target**;**

**3 输入框中只能输入数字**

项目中有些文本框是只允许输入数字字符的，如年龄、价格、数字编号等等。创建一个只允许输入数字的文本框，如下图-4所示：

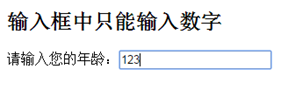


图-4

**参考答案**

此案例的难点在于考虑如下浏览器兼容性问题：

(1)禁止用户使用中文输入法可以使用CSS中的ime-mode:false来实现，但目前Chrome不支持此样式;

(2)可以为键盘按下事件添加处理方法，获取用户输入的字符键值，IE仅支持event.keyCode；而Firefox仅支持event.which;

(3)当用户输入的是非数字时，需阻止输入事件的发生，IE可以使用event.returnValue=false;而Firefox仅支持event.preventDefault();

(4)此外为了解决复制粘贴非数字字符，在按键弹起事件中，还应该对非数字字符进行剔除操作。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_03.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**输入框中只能输入数字**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript文件引用，创建页面基本元素**

可以使用CSS样式ime-mode:disabled，禁止在指定输入框中使用中文输入法；同时指定键盘按下和弹起两个事件的处理方法。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**输入框中只能输入数字**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript" **src=**"exercise05\_03.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<h2>输入框中只能输入数字</h2>**
10. **请输入您的年龄：<input style=**"ime-mode:disabled" **onkeypress=**"digitOnly(event)" **onkeyup=**"filterChar(this)"**/>**
11. **</**body**>**
12. **</**html**>**

**步骤三：创建外部JavaScript文件exercise05\_03.js**

分别为输入框的键盘按下事件创建digitOnly()函数和filterChar()函数。

1. //当按键按下时，阻止非数字字符的输入
2. **function** digitOnly**(**event**){**
3. **var** code **=** event**.**keyCode **||** event**.**which**;** //ie||firefox
4. **if((**code**<48)||(**code**>57)){**
5. **if(**event**.**preventDefault**){**    //firefox
6. event**.**preventDefault**();**
7. **}else{**                         //ie
8. event**.**returnValue **=** **false;**
9. **}**
10. **}**
11. **}**
12. //当按键弹起时，剔除输入中的非中文字符——解决复制粘贴进来的非数字
13. **function** filterChar**(**input**){**
14. input**.**value **=** input**.**value**.**replace**(**/[^0-9]/g**,**''**);**
15. **}**

**4 记录访问历史（提高题，选做）**

商城应用中，一般会记录用户最近浏览过的商品，即使浏览器关闭再次打开，仍可以看到自己上一次的浏览记录，从而便于用户查找自己之前浏览过的商品。

如图-5，用户第一次打开商城首页，提示“无浏览记录”：



图-5

当用户浏览过某商品之后，再次返回首页，可以看到浏览历史中已经有了该记录，如图-6和图-7所示：



图-6



图-7

***注意：***

***(1)浏览记录中若已经存在某商品，再次浏览该商品时，此商品会出现在“浏览历史”的最前端，后面相同的记录会被删除。***

***(2)历史记录不会无限增长，一般有一定的最大值，本案例中假设此值为6。***

***(3)本案例设定浏览记录最长可以保存30天。***



图-8

**参考答案**

记录浏览历史，在多个页面间传递数据，是cookie的典型应用场合。本案例的“首页”不仅要加载所有商品列表，还要试着从cookie中加载之前的浏览记录(该记录中只需要保存浏览过的商品的编号即可)；每个商品的“详情页面”，应该向cookie中添加一条浏览记录(即当前商品的编号)，只要用户再次回到“首页”，即可以显示出此次浏览记录。

“商品详情”页面中，将当前商品编号保存入cookie时，需要注意cookie的超时时间和访问路径问题——若不指定超时时间，浏览器一关闭这些cookie就会被销毁；若不指定访问路径，那“首页”必须与“详情页面”在同一个路径下或者“首页”在“详情页面”的子目录下，可能不符合应用的真实情景。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_04.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，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**记录访问历史**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部JavaScript和CSS文件引用，添加基本页面元素**

1. 本案例页面中只需要两个DIV即可，一个用于加载“所有商品列表”，另一个用于加载“浏览历史记录中的商品”，这两个DIV中的内容都是在页面加载完成时使用JavaScript加载进来的。
2. **<!**DOCTYPE html**>**
3. **<**html**>**
4. **<**head**>**
5. **<**title**>**记录访问历史**</**title**>**
6. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
7. **<link rel=**"stylesheet" **type=**"text/css" **href=**"exercise05\_04.css"**/>**
8. **<script type=**"text/javascript"**src=**"exercise05\_04.js"**></script>**
9. **</**head**>**
10. **<**body onload**=**"load(data)"**>**
11. **<p>网上书城-商品列表</p>**
12. **<hr/>**
13. **<div id=**"books"**></div>**
14. **<div class=**"clear"**></div>**
16. **<p>您曾经浏览过以下商品</p>**
17. **<hr/>**
18. **<div id=**"viewed"**></div>**
19. **</**body**>**
20. **</**html**>**

**步骤三：创建外部CSS文件exercise05\_04.css**

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

**步骤四：创建外部JavaScript文件exercise05\_04.js**

创建全局变量，模拟页面从服务器端接收到的待显示的数据：

1. //模拟从服务器获取到的具有特定格式的JSON数据
2. **var** data **=** **[** **{**
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. **}** **];**

创建页面加载完成后调用的函数load()，执行所有商品数据和浏览历史记录的加载：

1. /\*
2. \* 解析获取的JSON数据，每个书籍对象转换为一个DIV
3. \*/
4. **function** load**(**data**){**
5. //加载所有的书籍
6. loadAllBookList**(**data**);**
7. //加载已浏览的实际
8. loadViewedBookList**(**data**)**
9. **}**
10. /\*
11. \* 加载所有的书籍：使用文档片段封装所有新建的div
12. \*/
13. **function** loadAllBookList**(**data**){**
14. //将新创建的书籍放入一个文档代码片段
15. **var** fragment **=** document**.**createDocumentFragment**();**
16. **for(var** i**=0;** i**<**data**.**length**;** i**++){**
17. **var** div **=** createBookView**(**data**[**i**]);**
18. fragment**.**appendChild**(**div**);**
19. **}**
20. //加载所有的书籍：将文档片段对象添加到DOM树
21. document**.**getElementById**(**'books'**).**appendChild**(**fragment**);**
22. **}**
23. /\*
24. \* 读取cookie数据，根据其中保存的书籍编号，加载所有已浏览过的书籍
25. \* \*/
26. **function** loadViewedBookList**(**data**){**
27. **var** viewedBnos **=** getCookieValue**(**'viewedBnos'**);**
28. **if(**viewedBnos**){**    //存在浏览记录
29. **var** bnoArr **=** viewedBnos**.**split**(**','**);**
30. **var** fragment **=** document**.**createDocumentFragment**();**
31. **for(var** i**=0;** i**<**bnoArr**.**length**;** i**++){**    //遍历浏览过的所有书籍编号
32. **for(var** j**=0;** j**<**data**.**length**;** j**++){** //对于每一个书籍编号到全部书籍列表中查找对应的书籍对象
33. **if(**data**[**j**].**bno**==**bnoArr**[**i**]){**
34. **var** div **=** createBookView**(**data**[**j**]);**
35. fragment**.**appendChild**(**div**);**
36. **break;**
37. **}**
38. **}**
39. **}**
40. //将文档片段添加到DOM树
41. document**.**getElementById**(**'viewed'**).**appendChild**(**fragment**);**
42. **}else{**    //浏览记录为空
43. document**.**getElementById**(**'viewed'**).**innerHTML **=** '无浏览记录'**;**
44. **}**
45. **}**

上述方法需要把JSON数据中的一个书籍对象转换为一段HTML代码，创建需要的createBookView(book)函数：

1. /\*
2. \* 创建一个书籍对象对应的HTML元素
3. \* <div class="book">
4. \*         <img src="img"/>
5. \*         <a href="exercise05\_05\_id.html">bname</a>
6. \*         <span class="author">author</span>
7. \*         <span class="price">price</span>
8. \* </div>
9. \*/
10. **function** createBookView**(**book**){**
11. //每个书籍对象对应一个div
12. **var** div **=** document**.**createElement**(**'div'**);**
13. div**.**className **=** 'book'**;**
14. //创建书籍图片
15. **var** img **=** document**.**createElement**(**'img'**);**
16. img**.**src **=** book**.**img**;**
17. //创建书籍名称超链接
18. **var** a **=** document**.**createElement**(**'a'**);**
19. a**.**href **=** 'books/'**+**book**.**bno**+**".html"**;**
20. a**.**innerHTML **=** book**.**bname**;**
21. //创建显示作者的span
22. **var** spanAuthor **=** document**.**createElement**(**'span'**);**
23. spanAuthor**.**className **=** 'author'**;**
24. spanAuthor**.**innerHTML **=**book**.**author**;**
25. //创建显示price的span
26. **var** spanPrice **=** document**.**createElement**(**'span'**);**
27. spanPrice**.**className **=** 'price'**;**
28. spanPrice**.**innerHTML **=** '￥'**+**book**.**price**;**
30. div**.**appendChild**(**img**);**
31. div**.**appendChild**(**a**);**
32. div**.**appendChild**(**spanAuthor**);**
33. div**.**appendChild**(**spanPrice**);**
34. **return** div**;**
35. **}**

此外，loadViewedBookList()函数中还需要从document.cookie中读取cookie数据，获取保存在cookie中的已浏览过的商品的编号，故需要创建函数getCookieValue():

1. /\*为String类添加裁剪方法，cookie名中可能被浏览器添加空格\*/
2. String**.prototype.**trim **=** **function(){**
3. **return** **this.**replace**(**/(^\s\*)|(\s\*$)/g**,** ''**);**
4. **}**
5. /\*从所有cookie中查找具有指定cookieName的cookieValue\*/
6. **function** getCookieValue**(**cookieName**){**
7. //读取所有Cookie
8. **var** cookies **=** document**.**cookie**;**
9. **var** cookieValue **=** ''**;**
10. **if(**cookies**){**
11. **var** arr **=** cookies**.**split**(**';'**);** //用;拆分所有的cookie
12. **for(** **var** i**=0;** i**<**arr**.**length**;** i**++){**
13. **var** cookie **=** arr**[**i**];**        //键值对用=分割
14. **var** cookiePair **=** cookie**.**split**(** '=' **);**
15. **if(**cookiePair**[0].**trim**()==**cookieName**){**
16. cookieValue **=** cookiePair**[1];**
17. **break;**
18. **}**
19. **}**
20. **}**
21. **return** decodeURIComponent**(**cookieValue**);**
22. **}**

**步骤五：创建商品详情页面**

在当前目录下创建子目录books目录，其中创建八个测试HTML文件，分别命名为1001.html~1008.html，即每件商品都对应一个独立的HTML文件，文件名即为商品的编号。内容类似下面格式：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**商品详情**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript" src**=**"../exercise05\_04.js"**></**script**>**
7. **</**head**>**
8. **<**body onload**=**"writeCookie(1001)"**>**
9. **<**a href**=**"../exercise05\_04.html"**>**返回书籍列表**</**a**>**
10. **<**h2**>**商品详情**</**h2**>**
11. **<**img src**=**"1001detail.png"**/>**
12. **</**body**>**
13. **</**html**>**

***注意：***

***(1)上述商品详情页面使用的外部JavaScript文件与“首页”中使用的是同一个js文件，但因为不在同一个路径下，故需要使用../指定是上一级目录下的js文件;***

***(2)八个测试文件中body的onload事件处理方法调用的writeCookie()方法传递的参数即为商品的编号;***

***(3)处于简单考虑，此页面并未书写完整，每个商品的详情实际使用了一幅对应的图片，如1001号商品对应的详情图片为1001detail.png。***

**步骤六：修改外部JavaScript文件exercise05\_04.js，添加商品详情页面需要的writeCookie()方法**

1. /\*
2. \* 将指定商品编号写入客户端cookie，形如：
3. \* viewedBnos=1003,1008,1005;expires=xxxx;path=/
4. \*/
5. **function** writeCookie**(**bno**){**
6. **var** cookieName **=** 'viewedBnos'**;**
7. //从所有cookie中查找“浏览过的书籍编号”字符串，形如“viewedBnos=1008,1001,1009”
8. **var** cookieValue **=** getCookieValue**(**cookieName**);**
9. //添加新的bno到viewedBnos字符串，
10. //要求：    (1)将bno添加到viewedBnos最前面；若后面有重复的值，则删除
11. //     (2)viewedBnos中最多只保存6条记录
12. cookieValue **=** addNum**(**cookieValue**,** bno**,** **6);**
13. //重新把viewedBnos添加到document.cookie
14. **var** expires **=** **new** Date**();**
15. expires**.**setTime**(**expires**.**getTime**()+1000\*3600\*24\*30);**
16. **var** path **=** '/'**;**
17. //注意指定超时时间和访问路径——cookie默认只能在当前路径及子路径下访问
18. document**.**cookie **=** cookieName**+**'='**+**encodeURIComponent**(**cookieValue**)+**';expires='**+**expires**.**toGMTString**()+**";path="**+**path**;**
19. **}**
20. /\*
21. \* 参数str： String,形如“num1,num2,num3"
22. \* 参数num：Number类型
23. \* 参数maxCount：Integer,在data中可以保存的num的最多的数量
24. \* 将num添加到data的最前端，后面若有重复值则删除；且data中最多只保留maxCount个num,
25. \* 例如“1,3,5,2”再添加一个3后，返回“3,1,5,2”
26. \* \*/
27. **function** addNum**(**str**,**num**,** maxCount**){**
28. **var** arr **=** str**.**split**(**','**);**
29. //查找num是否已经存在，若存在则删除
30. **for(var** i**=0;** i**<**arr**.**length **&&** i**<**maxCount**-1;** i**++){**
31. **if(**arr**[**i**]==**num**){**
32. arr**.**splice**(**i**,1);**
33. **}**
34. **}**
35. //将待添加元素添加到数组头部
36. arr**.**unshift**(**num**);**
37. **return** arr**.**slice**(0,** maxCount**).**join**(**','**);**
38. **}**

至此，本案例全部实现，可以在Web服务器中部署项目文件并加以测试了。exercise05\_04.js中的完整内容如下所示：

1. //模拟从服务器获取到的具有特定格式的JSON数据
2. **var** data **=** **[** **{**
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. **}** **];**
59. /\*
60. \* 解析获取的JSON数据，每个书籍对象转换为一个DIV
61. \*/
62. **function** load**(**data**){**
63. //加载所有的书籍
64. loadAllBookList**(**data**);**
65. //加载已浏览的实际
66. loadViewedBookList**(**data**)**
67. **}**
68. /\*
69. \* 加载所有的书籍：使用文档片段封装所有新建的div
70. \*/
71. **function** loadAllBookList**(**data**){**
72. //将新创建的书籍放入一个文档代码片段
73. **var** fragment **=** document**.**createDocumentFragment**();**
74. **for(var** i**=0;** i**<**data**.**length**;** i**++){**
75. **var** div **=** createBookView**(**data**[**i**]);**
76. fragment**.**appendChild**(**div**);**
77. **}**
78. //加载所有的书籍：将文档片段对象添加到DOM树
79. document**.**getElementById**(**'books'**).**appendChild**(**fragment**);**
80. **}**
81. /\*
82. \* 读取cookie数据，根据其中保存的书籍编号，加载所有已浏览过的书籍
83. \* \*/
84. **function** loadViewedBookList**(**data**){**
85. **var** viewedBnos **=** getCookieValue**(**'viewedBnos'**);**
86. **if(**viewedBnos**){**    //存在浏览记录
87. **var** bnoArr **=** viewedBnos**.**split**(**','**);**
88. **var** fragment **=** document**.**createDocumentFragment**();**
89. **for(var** i**=0;** i**<**bnoArr**.**length**;** i**++){**    //遍历浏览过的所有书籍编号
90. **for(var** j**=0;** j**<**data**.**length**;** j**++){** //对于每一个书籍编号到全部书籍列表中查找对应的书籍对象
91. **if(**data**[**j**].**bno**==**bnoArr**[**i**]){**
92. **var** div **=** createBookView**(**data**[**j**]);**
93. fragment**.**appendChild**(**div**);**
94. **break;**
95. **}**
96. **}**
97. **}**
98. //将文档片段添加到DOM树
99. document**.**getElementById**(**'viewed'**).**appendChild**(**fragment**);**
100. **}else{**    //浏览记录为空
101. document**.**getElementById**(**'viewed'**).**innerHTML **=** '无浏览记录'**;**
102. **}**
103. **}**
104. /\*
105. \* 创建一个书籍对象对应的HTML元素
106. \* <div class="book">
107. \*         <img src="img"/>
108. \*         <a href="exercise05\_05\_id.html">bname</a>
109. \*         <span class="author">author</span>
110. \*         <span class="price">price</span>
111. \* </div>
112. \*/
113. **function** createBookView**(**book**){**
114. //每个书籍对象对应一个div
115. **var** div **=** document**.**createElement**(**'div'**);**
116. div**.**className **=** 'book'**;**
117. //创建书籍图片
118. **var** img **=** document**.**createElement**(**'img'**);**
119. img**.**src **=** book**.**img**;**
120. //创建书籍名称超链接
121. **var** a **=** document**.**createElement**(**'a'**);**
122. a**.**href **=** 'books/'**+**book**.**bno**+**".html"**;**
123. a**.**innerHTML **=** book**.**bname**;**
124. //创建显示作者的span
125. **var** spanAuthor **=** document**.**createElement**(**'span'**);**
126. spanAuthor**.**className **=** 'author'**;**
127. spanAuthor**.**innerHTML **=**book**.**author**;**
128. //创建显示price的span
129. **var** spanPrice **=** document**.**createElement**(**'span'**);**
130. spanPrice**.**className **=** 'price'**;**
131. spanPrice**.**innerHTML **=** '￥'**+**book**.**price**;**
133. div**.**appendChild**(**img**);**
134. div**.**appendChild**(**a**);**
135. div**.**appendChild**(**spanAuthor**);**
136. div**.**appendChild**(**spanPrice**);**
137. **return** div**;**
138. **}**
139. /\*为String类添加裁剪方法，cookie名中可能被浏览器添加空格\*/
140. String**.prototype.**trim **=** **function(){**
141. **return** **this.**replace**(**/(^\s\*)|(\s\*$)/g**,** ''**);**
142. **}**
143. /\*从所有cookie中查找具有指定cookieName的cookieValue\*/
144. **function** getCookieValue**(**cookieName**){**
145. //读取所有Cookie
146. **var** cookies **=** document**.**cookie**;**
147. **var** cookieValue **=** ''**;**
148. **if(**cookies**){**
149. **var** arr **=** cookies**.**split**(**';'**);** //用;拆分所有的cookie
150. **for(** **var** i**=0;** i**<**arr**.**length**;** i**++){**
151. **var** cookie **=** arr**[**i**];**        //键值对用=分割
152. **var** cookiePair **=** cookie**.**split**(** '=' **);**
153. **if(**cookiePair**[0].**trim**()==**cookieName**){**
154. cookieValue **=** cookiePair**[1];**
155. **break;**
156. **}**
157. **}**
158. **}**
159. **return** decodeURIComponent**(**cookieValue**);**
160. **}**
161. /\*
162. \* 将指定商品编号写入客户端cookie，形如：
163. \* viewedBnos=1003,1008,1005;expires=xxxx;path=/
164. \*/
165. **function** writeCookie**(**bno**){**
166. **var** cookieName **=** 'viewedBnos'**;**
167. //从所有cookie中查找“浏览过的书籍编号”字符串，形如“viewedBnos=1008,1001,1009”
168. **var** cookieValue **=** getCookieValue**(**cookieName**);**
169. //添加新的bno到viewedBnos字符串，
170. //要求：    (1)将bno添加到viewedBnos最前面；若后面有重复的值，则删除
171. //     (2)viewedBnos中最多只保存6条记录
172. cookieValue **=** addNum**(**cookieValue**,** bno**,** **6);**
173. //重新把viewedBnos添加到document.cookie
174. **var** expires **=** **new** Date**();**
175. expires**.**setTime**(**expires**.**getTime**()+1000\*3600\*24\*30);**
176. **var** path **=** '/'**;**
177. //注意指定超时时间和访问路径——cookie默认只能在当前路径及子路径下访问
178. document**.**cookie **=** cookieName**+**'='**+**encodeURIComponent**(**cookieValue**)+**';expires='**+**expires**.**toGMTString**()+**";path="**+**path**;**
179. **}**
180. /\*
181. \* 参数str： String,形如“num1,num2,num3"
182. \* 参数num：Number类型
183. \* 参数maxCount：Integer,在data中可以保存的num的最多的数量
184. \* 将num添加到data的最前端，后面若有重复值则删除；且data中最多只保留maxCount个num,
185. \* 例如“1,3,5,2”再添加一个3后，返回“3,1,5,2”
186. \* \*/
187. **function** addNum**(**str**,**num**,** maxCount**){**
188. **var** arr **=** str**.**split**(**','**);**
189. //查找num是否已经存在，若存在则删除
190. **for(var** i**=0;** i**<**arr**.**length **&&** i**<**maxCount**-1;** i**++){**
191. **if(**arr**[**i**]==**num**){**
192. arr**.**splice**(**i**,1);**
193. **}**
194. **}**
195. //将待添加元素添加到数组头部
196. arr**.**unshift**(**num**);**
197. **return** arr**.**slice**(0,** maxCount**).**join**(**','**);**
198. **}**