**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. **}**